Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 8(2): 3561–3605.
DOI: 10.3934/math.2023181
Received: 06 September 2022
Revised: 24 October 2022
Accepted: 09 November 2022
Published: 23 November 2022

Research article

A direct integral pseudospectral method for solving a class of infinite-horizon optimal control problems using Gegenbauer polynomials and certain parametric maps

Kareem T. Elgindy[1,2,*] **and Hareth M. Refat**[3]

[1] Mathematics Department, College of Computing and Mathematics, King Fahd University of Petroleum & Minerals, Dhahran 31261, Kingdom of Saudi Arabia

[2] IRC for Membrances & Water Security, King Fahd University of Petroleum & Minerals, Dhahran 31261, Kingdom of Saudi Arabia

[3] Mathematics Department, Faculty of Science, Sohag University, Sohag 82524, Egypt

* **Correspondence:** Email: kareem.elgindy@kfupm.edu.sa, kareem.elgindy@gmail.com;
Tel: +966591398575.

Abstract: We present a novel direct integral pseudospectral (PS) method (a direct IPS method) for solving a class of continuous-time infinite-horizon optimal control problems (IHOCs). The method transforms the IHOCs into finite-horizon optimal control problems in their integral forms by means of certain parametric mappings, which are then approximated by finite-dimensional nonlinear programming problems (NLPs) through rational collocations based on Gegenbauer polynomials and Gegenbauer-Gauss-Radau (GGR) points. The paper also analyzes the interplay between the parametric maps, barycentric rational collocations based on Gegenbauer polynomials and GGR points and the convergence properties of the collocated solutions for IHOCs. Some novel formulas for the construction of the rational interpolation weights and the GGR-based integration and differentiation matrices in barycentric-trigonometric forms are derived. A rigorous study on the error and convergence of the proposed method is presented. A stability analysis based on the Lebesgue constant for GGR-based rational interpolation is investigated. Two easy-to-implement pseudocodes of computational algorithms for computing the barycentric-trigonometric rational weights are described. Three illustrative test examples are presented to support the theoretical results. We show that the proposed collocation method leveraged with a fast and accurate NLP solver converges exponentially to near-optimal approximations for a coarse collocation mesh grid size. The paper also shows that typical direct spectral/PS and IPS methods based on classical Jacobi polynomials and certain parametric maps usually diverge as the number of collocation points grow large if the computations are carried out using floating-point arithmetic and the discretizations use a single mesh grid, regardless of whether they are of Gauss/Gauss-Radau type or equally spaced.

Keywords: algebraic map; Gauss-Radau points; Gegenbauer polynomials; infinite horizon; integration matrix; logarithmic map; optimal control; pseudospectral method
Mathematics Subject Classification: 65D05, 65D15, 65D30, 65D32

1. Introduction

Arguably, one of the most impactful numerical methods for solving continuous-time optimal control problems (CTOCPs) in the 20th century has been direct pseudospectral (PS) methods, which can accurately reduce CTOCPs into optimization problems of standard forms that can be easily treated by using typical optimization methods. The key success of these methods lies in their ability to converge to sufficiently smooth solutions with exponential rates by using relatively coarse mesh grids. PS methods are considered to be "one of the biggest technologies for solving PDEs" that were largely developed about a half century ago since the pioneering works of Orszag [1] and Patterson Jr and Orszag [2]. They have been continuously refined and extended in later decades to solve many problems in various scientific areas that were only tractable by these techniques. Perhaps, one of the brightest moments in the course of their development appeared on March 3, 2007, when the International Space Station completed a 180-degree maneuver without using any propellant by tracking an attitude trajectory developed with PS optimal control theory, saving NASA one million USD [3]. The development of PS methods continues to be very active, and the progress in this research area has been remarkable in recent years; see the works in [4–6], for a few references on the solution of optimal control problems using various schemes of PS methods.

PS methods are closely related to the popular class of spectral methods, but they expand the solutions in terms of their grid point values by means of interpolation in lieu of global and, usually, orthogonal basis polynomials. Such a nodal representation is extremely useful in the sense that the solution values are immediately available at the collocation points once the full discretization is implemented, as the the governing equations are satisfied pointwise in the physical space, whereas modal representations require a further step of computing the modal approximation after calculating the coefficients of the basis function expansions [7]. This places PS methods at the front of highly accurate methods that are particularly easy to apply to equations with variable coefficients and nonlinearities [8]. Clear expositions of spectral and PS methods exhibiting a wide range of outlooks on the subject include, to mention a few, the books [9–11].

Two of the most common alternatives to direct spectral and PS methods are indirect methods and parameterization methods, which include control parameterization and state and control parameterization. As their names suggest, only the control variable is parameterized in a control parameterization method, and the differential equations are solved via numerical integration, while both state and control variables are parameterized in state and control parameterization methods, and the differential equations are converted into algebraic constraints. Typical examples of control parameterization methods include shooting methods and multiple shooting methods. On the other hand, an indirect method requires first the derivation of the often complicated first-order necessary optimality conditions, which include the adjoint equations, the control equations and the transversality conditions, before carrying out the numerical discretization. In fact, direct PS methods often present more computational efficiency and robustness over these classical methods on solving optimal control problems due to many reasons; to mention a few, (i) multiple shooting methods do not handle problems with singular arcs appropriately without a priori information on the structure of the trajectories [12]; (ii) direct shooting methods are often intensive computationally and associated with sensitivity issues; in particular, the ability to successfully use a direct shooting method declines as the number of variables grows large [13]; (iii) the boundary value problem resulting from the necessary conditions of optimality are extremely sensitive to initial guesses in indirect methods [14]; (iv) in contrast to indirect methods and direct shooting methods, direct PS methods do not require a priori knowledge of the active and inactive arcs for problems with inequality path constraints [15]; (v) the user does not have to be concerned with the adjoint variables or the switching structures to determine the optimal control in direct PS methods [16]; (vi) direct PS methods show much bigger convergence radii than either the indirect methods or direct shooting methods, as they are much less sensitive to the initial guesses [17]; (vii) direct PS methods are often memory minimizing when performed using orthogonal collocation approximations, as they result in finite-dimensional nonlinear programming problems (NLPs) with considerably lower dimensions compared to other competitive methods in the literature [18]; (viii) direct PS methods perform numerical differentiation through constant operators that can be stored for certain sets of collocation points, and invoked later when implementing the computational algorithms; and (ix) there is no need for the Lyapunov function to construct the asymptotically stabilizing control [6]. The reader may consult [19] for a further review on spectral and PS methods and their advantages against other classical methods.

A robust variant of PS methods is the class of integral PS (IPS) methods (that is, PS integration methods), which is closely related to PS methods, but it requires an initial step of reformulating the dynamical system equations in their integral form first before the collocation phase starts; thus, it avoids the degradation of precision that is often caused by numerical differentiation processes [20,21]. The integral reformulation can be performed by either a direct integration of the dynamical system equations if they have constant coefficients, or by approximating the solution's highest-order derivative involved in the problem by using a nodal finite series in terms of its grid point values and then solving for those grid point values before successively integrating back in a stable manner to obtain the sought solution grid point values. It has been shown in recent decades through a large number of publications that IPS methods often exhibit faster convergence rates, produce higher accuracy and are much less sensitive to the types of collocation points used than the usual PS methods; see [22–24] to mention a few. Historically, the spectral approximation of the integral form of differential equations was put forward in the 1960s by Clenshaw and Curtis [25] in the spectral space, and by El-Gendi [26] in the physical space.

Among the many classes of CTOCPs, infinite-horizon optimal control problems (IHOCs) and optimal control problems defined on sufficiently large intervals have attracted a lot of research interest due to the extent of their applications in economics, engineering, computer science, business and management science, bio-medicine, aerospace, energy, etc.; see [27–29] to mention a few. Some classical results on the existence of solutions for IHOCs can be found in [30–33]. One of the most general and well-known results on the existence of solutions to IHOCs was proved by Balder [34] using the notion of uniform integrability. Sufficient conditions for the existence of a finitely optimal solution for a class of nonlinear IHOCs were derived by Carlson [35] under minimal convexity and seminormality conditions. An existence and uniqueness theorem for a class of IHOCs was proved by

Wang [36] under certain conditions. Existence and uniqueness results for a class of linear-quadratic, convex IHOCs in weighted Sobolev spaces for the state and weighted Lebesgue spaces for the control were obtained by Pickenhain [37]. A recent extension to the existence results of Balder [34] to the case in which the integral functional is understood as an improper integral was proved by Besov [38] using the notion of uniform boundedness of pieces of the objective functional that was proposed earlier by Dmitruk and Kuz'kina [39]. Aseev [40] derived some sufficient conditions for the existence and boundedness of optimal controls for a class of generally nonlinear IHOCs without necessarily having a bounded set of control constraints. Basco and Frankowska [41] obtained some existence and uniqueness results for weak solutions of the nonautonomous Hamilton–Jacobi–Bellman equation associated with a class of IHOCs for the class of lower semicontinuous functions vanishing at infinity and under certain conditions of controllability. The most important and well-known necessary conditions of optimality were first derived by Halkin [42]; see also [33, Theorem 2.3].

While many direct PS methods appeared in the literature for solving finite-horizon optimal control problems (FHOCs), we could only find a few works on deterministic IHOCs governed by integer-order differential equations using this class of methods. In particular, we recognize the Legendre-Gauss (LG) and Legendre-Gauss-Radau (LGR) PS methods in [43–45], the flipped Legendre-Gauss-Radau PS method (FRPM) in [46] and the transformed Legendre spectral method in [47]. Although Legendre polynomials are commonly used in PS methods designed to solve IHOCs, we shall explore in our work the possibility of whether we could achieve better accuracy and convergence rates using Gegenbauer polynomials (that is, ultraspherical polynomials). There are a number of reasons that prompt us to consider this family of polynomials as a viable alternative to perform discretizations of IHOCs. (i) First, observe that Gegenbauer polynomials include both Chebyshev and Legendre polynomials as part of its bigger family, so all theoretical and experimental results on Gegenbauer polynomials directly apply to Chebyshev and Legendre polynomials by definition. (ii) Being a part of Gegenbauer polynomials allows us to apply any of the Chebyshev and Legendre polynomials with a single selection of the Gegenbauer parameter (index) α, as one can simply set $\alpha = 0$ or $1/2$ in your code, thus giving us more flexibility. (iii) Gegenbauer polynomials are very useful in eliminating the Gibbs phenomenon and recovering the spectral accuracy up to the discontinuity points [48–50]. (iv) One measure for assessing the quality of spectral and PS methods in numerical discretizations is concerned with how large is the number of terms that are required in a spectral/PS expansion to achieve a certain level of accuracy. Of course, the smaller the number of terms, the more efficient the method is in terms of speed and computational complexity. In applications like CTOCPs, this property leads to optimization problems on the small scale, which can be solved very quickly with reduced computational work at a concrete level by using modern optimization software [18, 21, 23]. Now, with this being mentioned, it is important to realize that Chebyshev and Legendre polynomials are usually optimal for large spectral/PS expansions under the Chebyshev and Euclidean norms, respectively, but they are not necessarily optimal for a small/medium range; this is an observation that was proven numerically in a number of papers for certain polynomial and rational interpolations and collocations; see [8, 20, 51] to mention a few, which give us another reason to apply Gegenbauer polynomials as proper basis polynomials that may provide faster convergence rates. (v) A stability analysis conducted in [24] and grounded in the Lebesgue constant for polynomial interpolations in Lagrange-basis form based on flipped Gegenbauer-Gauss-Radau (FGGR) points showed that the Lebesgue constant is not minimal for

Chebyshev polynomials, but, rather, was minimal for Gegenbauer polynomials associated with negative α values. This analysis proved, with no doubt, that some Gegenbauer polynomials with negative α values could be more plausible for employment in basis-form polynomial interpolation/collocation for short-medium-range mesh grid sizes. This observation is consistent with the earlier work of Light [52], who proved in the late 1970's that the Chebyshev and Legendre projection operators cannot be minimal as the norms of Gegenbauer projection operators increase monotonically with α for small expansions.

In light of the above arguments, we were motivated in this work to develop a novel direct IPS method for solving IHOCs using Gegenbauer polynomials, and to study its convergence. To this end, we derive some accurate and numerically stable Gegenbauer-Gauss-Radau (GGR)-based rational interpolation formulas and describe two useful computational algorithms for constructing their barycentric weights. We also show how to derive the associated quadrature formulas required for numerical integration in time. We shall then use these numerical instruments to approximate the optimal state and control variables after transforming the IHOC into an FHOC in integral form (FHOCI) by means of certain parametric maps and rational collocation. During the course of our paper presentation, we shall also try to investigate a number of interesting relevant questions to our work. For instance, which parametric map is more suitable for GGR-based rational collocations? How should we choose the Gegenbauer parameter values to carry out collocations in practice? A "poor" choice of α can largely ruin the accuracy of the numerical scheme, while a "good" choice can, in many cases, furnish superb approximations with higher accuracy than those enjoyed by Chebyshev and Legendre polynomials for sufficiently smooth functions by using relatively coarse mesh grids. Do PS and IPS methods based on Chebyshev, Legendre and Gegenbauer polynomials generally converge to the solutions of IHOCs for large mesh sizes? If they do not, then what are the causes? Through rigorous stability, error and convergence analyses, we shall prove that such methods often converge at an exponential rate to near exact solutions by using relatively small mesh grids, but they usually diverge for fine meshes under certain parametric maps.

The proposed method inherits all of the merits of the direct IPS method stated above; in addition, the current paper presents the following novel contributions to the literature of IHOCPs: (i) For a coarse collocation mesh grid size under certain parametric maps, we show that direct IPS methods often converge exponentially to near-optimal solutions of IHOCs when leveraged with a fast and accurate NLP solver. (ii) Despite the exceedingly accurate approximations achieved at coarse meshes, we prove that direct IPS methods based on Gegenbauer polynomials and certain parametric maps usually diverge as the number of collocation points grow large if the computations are carried out using floating-point arithmetic and the discretizations use a single mesh grid, regardless of whether they are of Gauss/Gauss-Radau (GR) type or equally spaced, which is a result that can be readily extended to include classical Jacobi polynomials in general. (iii) We derive some novel formulas for the construction of GGR-based rational interpolation weights, leading to a more numerically stable interpolation procedure, which we prefer to call "the switched rational (SR) interpolation," due to the switching nature of the novel algorithm we developed to accurately calculate the barycentric weights. (iv) More novel formulas for the construction of GGR-based integration and differentiation matrices in barycentric-trigonometric forms are derived. (v) We show that Legendre polynomials are (near) optimal basis polynomials for GGR-based SR collocations over coarse meshes. (vi) We show that Gegenbauer polynomials associated with certain nonnegative index values are more apt for

GGR-based SR interpolations over fine meshes. (vii) We prove that parametric logarithmic maps are more apt for the domain transformation of IHOCs than parametric algebraic maps for GGR-based SR collocations. (viii) To the best of our knowledge, this paper introduces the first direct IPS method for solving IHOCs using Gegenbauer polynomials and algebraic-logarithmic parametric maps.

The rest of the article is organized as follows. Sections 2 and 3 describe the IHOC under study and its transformation into an FHOCI via various parametric maps. Section 4 presents the discretization scheme of the FHOCI passing through the construction of the needed barycentric rational interpolants and their quadratures, and it closes with a setup of the IPS rational collocation at the GGR points and the optimality necessary conditions of the obtained NLP in Sections 4.1–4.3. Section 4.1 includes an analysis of the stability and sensitivity of the GGR-based rational interpolation/collocation developed in this paper. Rigorous error and convergence analyses are conducted in Section 5. Some divergence results of typical IPS collocation schemes of the FHOCI for fine meshes of the Gauss type using certain parametric maps are derived in Section 5. Simulation results are shown in Section 6, followed by some conclusions and future works in Section 7. The derivation of the barycentric rational formulas necessary for constructing the GGR-based differentiation matrix is shown in Appendix A. Two easy-to-implement pseudocodes of computational algorithms for computing the barycentric weights of our new SR interpolation method are described in Appendix B.

2. Problem statement

Consider the following nonlinear, autonomous control system of ordinary differential equations:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad t \in [0, \infty), \tag{2.1}$$

which is subject to the following system of initial conditions:

$$\boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{2.2}$$

where $\boldsymbol{x}(t) = [x_1(t), x_2(t), \ldots, x_{n_x}(t)]^\top \in \mathbb{R}^{n_x}, \boldsymbol{u}(t) = [u_1(t), u_2(t), \ldots, u_{n_u}(t)]^\top \in \mathbb{R}^{n_u}, \boldsymbol{x}_0 = [x_{1,0}, x_{2,0}, \ldots, x_{n_x,0}]^\top \in \mathbb{R}^{n_x}$ is a constant specified vector, and $\boldsymbol{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x} : \boldsymbol{f} = [f_1, f_2, \ldots, f_{n_x}]^\top$. The problem is to find the optimal control $\boldsymbol{u}(t)$ and the corresponding state trajectory $\boldsymbol{x}(t)$ on the semi-infinite-domain $[0, \infty)$ that satisfy Eqs (2.1) and (2.2) while minimizing the functional

$$J = \int_0^\infty g(\boldsymbol{x}(t), \boldsymbol{u}(t))\, dt, \tag{2.3}$$

where $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$. We assume that $\boldsymbol{f}$ and g are generally nonlinear, continuously differentiable functions with respect to their arguments, and that the nonlinear IHOC (2.1)–(2.3) has a unique solution. In the rest of the article, for any row/column vector $\boldsymbol{y} = (y_i)_{1 \le i \le n}$ with $y_i \in \mathbb{R}\, \forall i$ and real-valued function $h : \boldsymbol{\Omega} \subseteq \mathbb{R} \to \mathbb{R}$, the notation $h(\boldsymbol{y})$ stands for a vector of the same size and structure of $\boldsymbol{y}$ such that $h(y_i)$ is the ith element of $h(\boldsymbol{y})$. Moreover, by $\boldsymbol{h}(\boldsymbol{y})$, we mean $[h_1(\boldsymbol{y}), \ldots, h_m(\boldsymbol{y})]^\top$ for any m-dimensional column vector function $\boldsymbol{h}$, with the realization that the definition of each array $h_i(\boldsymbol{y})$ follows the former notation rule for each i.

3. Transformation of the IHOC into an FHOC

Given a differentiable, strictly monotonic mapping $T : [0, \infty) \to [-1, 1)$ defined by $T(\tau) = t$, one can transform the IHOC (2.1)–(2.3) into the following FHOC:

$$\min J = \int_{-1}^{1} T'(\tau) g\left(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau)\right) d\tau, \tag{3.1a}$$

subject to

$$\dot{\tilde{\boldsymbol{x}}}(\tau) = T'(\tau)\boldsymbol{f}(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau)), \quad \tau \in [-1, 1), \tag{3.1b}$$

$$\tilde{\boldsymbol{x}}(-1) = \boldsymbol{x}_0, \tag{3.1c}$$

where $\boldsymbol{f}\left(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau)\right) = \left[f_1\left(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau)\right), \ldots, f_{n_x}\left(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau)\right)\right]^{\top}$ and $\tilde{\boldsymbol{\eta}}(\tau) = \boldsymbol{\eta}\left(T(\tau)\right) \,\forall \boldsymbol{\eta} \in \{\boldsymbol{x}, \boldsymbol{u}\}$. To take advantage later of the well-conditioning of numerical integration operators during the collocation phase, we rewrite Eq (3.1b) in its integral formulation, as follows:

$$\tilde{\boldsymbol{x}}(\tau) = \int_{-1}^{\tau} T'(z)\boldsymbol{f}\left(\tilde{\boldsymbol{x}}(z), \tilde{\boldsymbol{u}}(z)\right) dz + \boldsymbol{x}_0, \quad \tau \in [-1, 1). \tag{3.1d}$$

We refer to the FHOC described by Eqs (3.1a), (3.1c) and (3.1d) by the FHOCI. A wide variety of defining formulas exist for the mapping T. Five common defining formulas of such a mapping that occurred in the literature are as follows:

$$T_{1,L}(\tau) = \frac{L(1+\tau)}{1-\tau}, \quad ([53]) \tag{3.2a}$$

$$T_{2,L}(\tau) = L \ln\left(\frac{2}{1-\tau}\right), \quad ([54]) \tag{3.2b}$$

$$\varphi_a(\tau) = \frac{1+\tau}{1-\tau}, \quad ([55]) \tag{3.2c}$$

$$\left.\begin{aligned} \varphi_b(\tau) &= \ln\left(\frac{2}{1-\tau}\right), \\ \varphi_c(\tau) &= \ln\left(\frac{4}{(1-\tau)^2}\right), \end{aligned}\right\} \quad ([12]) \tag{3.2d}$$

where $L \in \mathbb{R}^+$ is a scaling parameter that can stretch the image of the interval $[-1, 1]$ in the codomain $[0, \infty)$ as desired. An "optimal" choice of L value can significantly improve the quality of the discrete approximations, as we shall demonstrate later in Section 6. We refer to L as "the map scaling parameter." The parametric maps $T_{i,L}, i = 1, 2$ are often referred to as "algebraic" and "logarithmic" maps in the literature, respectively. Notice that the maps defined by Eqs (3.2c) and (3.2d) are special cases of the parametric maps $T_{i,L}, i = 1, 2$; in particular, $T_{1,1} = \varphi_a, T_{2,1} = \varphi_b$ and $T_{2,2} = \varphi_c$. Since the value of either parametric map varies when α varies for arguments of type GGR points, it is more convenient in this work to denote them by $T_{i,L}^{(\alpha)}, i = 1, 2$ to emphasize this fact. In particular, we prefer to generalize Eqs (3.2a) and (3.2b) to implicitly allow for maps of a wide spectrum of α values, as follows:

$$T_{1,L}^{(\alpha)}(\tau) = \frac{L(1+\tau)}{1-\tau}, \tag{3.3a}$$

$$T_{2,L}^{(\alpha)}(\tau) = L\ln\left(\frac{2}{1-\tau}\right). \tag{3.3b}$$

Mesh-like surfaces of both parametric mappings are shown in Figures 1 and 2 for several values of n, L and α. Both figures show that the parametric mappings (i) increase monotonically for decreasing values of α while holding n and L fixed, (ii) increase monotonically for increasing values of L while holding n and α fixed and (iii) increase monotonically for increasing values of n while holding L and α fixed. Moreover, near $\tau = 1$, the rate of increase of $T_{1,L}^{(\alpha)}$ with respect to any of the arguments n, L and α while holding the others fixed is much larger than that of $T_{2,L}^{(\alpha)}$, which grows very slowly. Loosely put, the stretching of the mesh grid near $\tau = 1$ is stronger for $T_{1,L}^{(\alpha)}$ than $T_{2,L}^{(\alpha)}$.

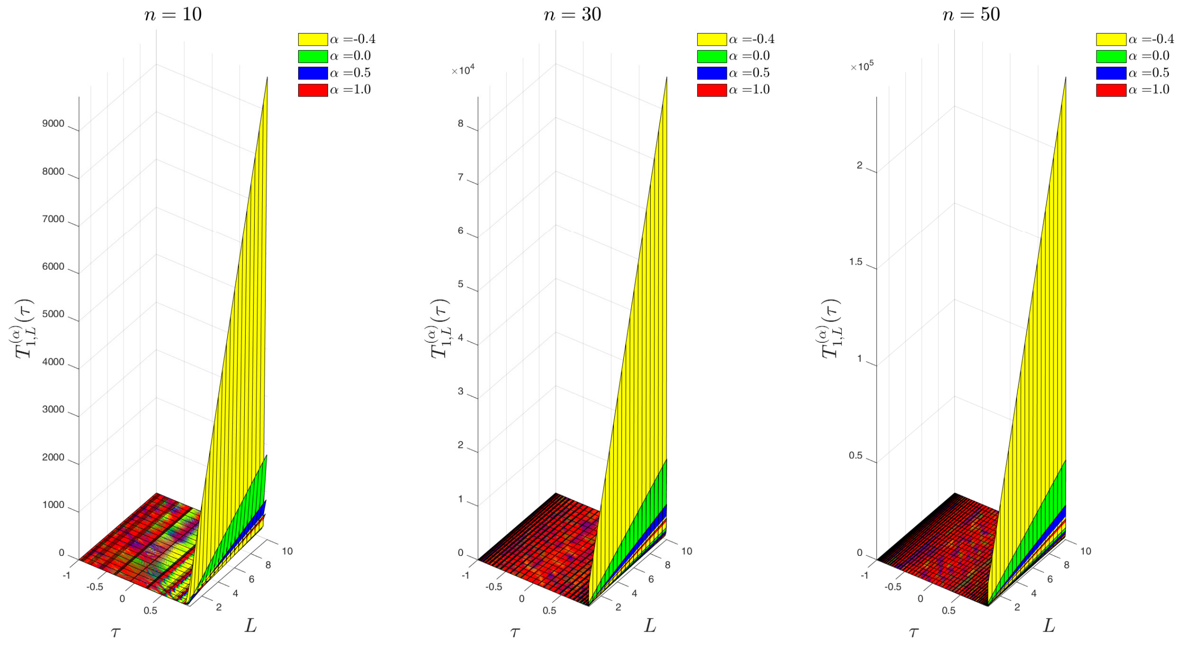

Figure 1. Mesh-like surfaces of the parametric mapping $T_{1,L}^{(\alpha)}$ on the discrete rectangular domain $\mathbf{\Omega}_n = \{(\tau_i, L) : L = 0.5(0.5)10, i = 0,\ldots,n\}$ for $n = 10(20)50$ and $\alpha = -0.4, 0, 0.5, 1$.

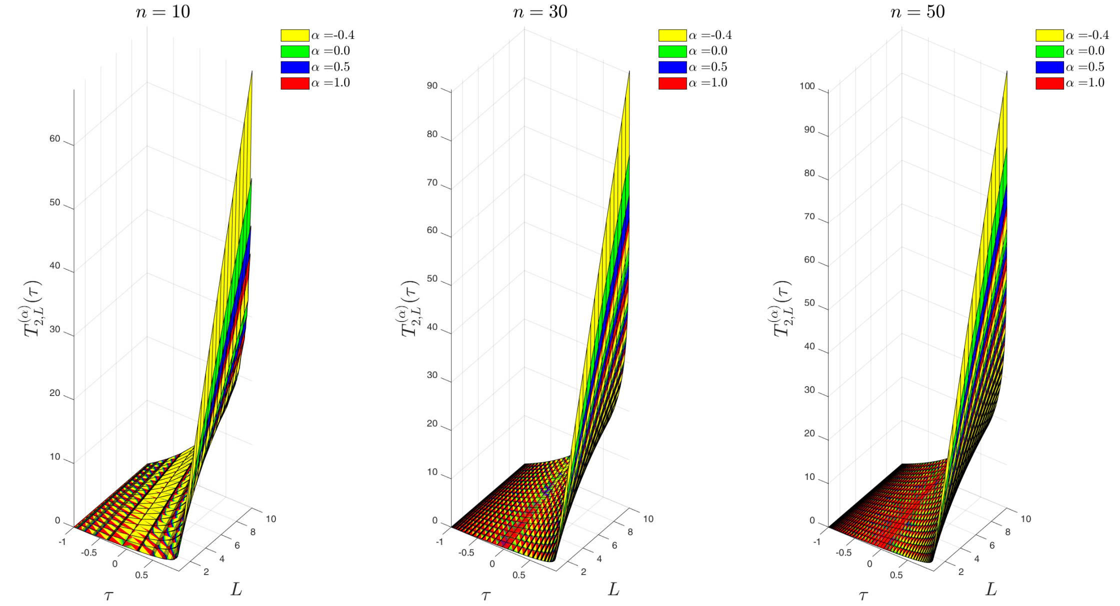

Figure 2. Mesh-like surfaces of the parametric mapping $T_{2,L}^{(\alpha)}$ on the discrete rectangular domain $\mathbf{\Omega}_n = \{(\tau_i, L) : L = 0.5(0.5)10, i = 0, \ldots, n\}$ for $n = 10(20)50$ and $\alpha = -0.4, 0, 0.5, 1$.

4. Numerical discretization of the FHOCI

In this section, we provide a description of the proposed numerical discretization of the FHOCI using an IPS method based on Gegenbauer polynomials and GGR points.

4.1. Barycentric rational interpolation at the GGR points

Let $\mathbb{Z}^+$ be the set of positive integers, $G_n^{(\alpha)}(\tau)$ be the nth-degree Gegenbauer polynomial with $\alpha > -1/2$ and $G_n^{(\alpha)}(1) = 1$, $\forall n \in \mathbb{Z}_0^+ = \mathbb{Z}^+ \cup \{0\}$, and $\mathbb{S}_n = \left\{\tau_k : \mathcal{G}_{n+1}^{(\alpha)}(\tau_k) = 0 \text{ for } k = 0, \ldots, n, \text{ and } -1 = \tau_0 < \tau_1 < \ldots < \tau_n\right\}$ be the set of GGR nodes, where $\mathcal{G}_{n+1}^{(\alpha)}(\tau) = G_n^{(\alpha)}(\tau) + G_{n+1}^{(\alpha)}(\tau)$. The orthonormal Gegenbauer basis polynomials are defined by $\phi_j^{(\alpha)}(\tau) = G_j^{(\alpha)}(\tau)/\sqrt{\lambda_j}$, where

$$\lambda_j = \frac{2^{2\alpha-1} j! \Gamma^2(\alpha + \frac{1}{2})}{(j+\alpha)\Gamma(j+2\alpha)}, \quad j = 0, \ldots, n. \tag{4.1}$$

They satisfy the discrete orthonormality relation

$$\sum_{j=0}^{n} \varpi_j \phi_s^{(\alpha)}(\tau_j) \phi_k^{(\alpha)}(\tau_j) = \delta_{sk}, \quad s, k = 0, \ldots, n, \tag{4.2}$$

where ϖ_j, $j = 0, 1, \ldots, n$, are the corresponding Christoffel numbers of the GGR quadrature formula on the interval $[-1, 1]$ defined by

$$\varpi_0 = \left(\alpha + \frac{1}{2}\right) \vartheta_0, \tag{4.3a}$$

$$\varpi_j = \vartheta_j, \quad j = 1,2,\ldots,n, \tag{4.3b}$$

with

$$\vartheta_j = 2^{2\alpha-1} \frac{\Gamma^2\left(\alpha+\frac{1}{2}\right) n!}{\left(n+\alpha+\frac{1}{2}\right)\Gamma(n+2\alpha+1)} \left(1-\tau_j\right)\left(G_n^{(\alpha)}(\tau_j)\right)^{-2}, \quad j = 0,1,\ldots,n. \tag{4.3c}$$

Given a set of $n+1$ data points $\{(\tau_i, f_i)\}_{i=0}^n$, the Gegenbauer polynomial interpolant $P_n f$ in Lagrange form is defined by

$$P_n f(\tau) = \sum_{i=0}^{n} f_i \mathcal{L}_{n,i}(\tau), \tag{4.4}$$

where $\mathcal{L}_{n,i}$ are the Lagrange polynomials given by

$$\mathcal{L}_{n,i}(\tau) = \frac{\prod_{k\neq i}(\tau-\tau_k)}{\prod_{k\neq i}(\tau_i-\tau_k)}, \quad \forall i. \tag{4.5}$$

$P_n f$ can be evaluated fast and more stably by evaluating Lagrange polynomials through the "true" barycentric formula

$$\mathcal{L}_{n,i}(\tau) = \frac{\xi_i}{\tau-\tau_i} \Big/ \sum_{j=0}^{n} \frac{\xi_j}{\tau-\tau_j}, \quad \forall i, \tag{4.6}$$

which brings into play the barycentric weights $\xi_i, i = 0,\ldots,n$, given by

$$\xi_i = \frac{1}{\prod_{i\neq j}^{n}\left(\tau_j-\tau_i\right)}, \quad \forall i. \tag{4.7}$$

An interpolation in Lagrange form with Lagrange polynomials defined by Eq (4.6) is often referred to as "a barycentric rational interpolation." The barycentric weights associated with the GGR points can be expressed explicitly in terms of the corresponding Christoffel numbers through the following theorem.

Theorem 4.1. *The barycentric weights for the GGR points are given by*

$$\xi_0 = -\sqrt{(2\alpha+1)\varpi_0}, \tag{4.8a}$$

$$\xi_i = (-1)^{i-1}\sqrt{(1-\tau_i)\,\varpi_i}, \quad i = 1,2,\ldots,n. \tag{4.8b}$$

Proof. Let $P_n^{(\alpha,\beta)}(\tau)$ be the Jacobi polynomial of degree n and associated with the parameters $\alpha,\beta > -1$, as normalized by Szegö [56]. Through [56, Eq (4.5.4)] and [18, Eq (A.1)], we have

$$\begin{aligned}
(1+\tau)P_n^{(\alpha-1/2,\alpha+1/2)}(\tau) &= \frac{2}{2n+2\alpha+1}\left[(n+\alpha+1/2)\,P_n^{(\alpha-1/2,\alpha-1/2)}(\tau) + (n+1)\,P_{n+1}^{(\alpha-1/2,\alpha-1/2)}(\tau)\right] \\
&= \frac{2}{2n+2\alpha+1}\left[\frac{(n+\alpha+1/2)\,\Gamma(n+\alpha+1/2)}{n!\,\Gamma(\alpha+1/2)} G_n^{(\alpha)}(\tau) + \frac{(n+1)\,\Gamma(n+\alpha+3/2)}{(n+1)!\,\Gamma(\alpha+1/2)} G_{n+1}^{(\alpha)}(\tau)\right] \\
&= \frac{\Gamma(n+\alpha+1/2)}{n!\,\Gamma(\alpha+1/2)} \mathcal{G}_{n+1}^{(\alpha)}(\tau).
\end{aligned} \tag{4.9}$$

Therefore, $(1+\tau_i)P_{n+1}^{(\alpha-1/2,\alpha+1/2)}(\tau_i) = \mathcal{G}_{n+1}^{(\alpha)}(\tau_i)\;\forall i$, and Eqs (4.8a) and (4.8b) can be derived from [57, Theorem 3.6] by replacing both α and β with $\alpha - 1/2$. □

Recall that the GGR points cluster near ±1 as $n \to \infty$, so Eq (4.8b) may suffer from cancellation errors for values of τ_i sufficiently close to 1. The following theorem provides two alternative trigonometric forms of Eq (4.8b).

Theorem 4.2. *The barycentric weights corresponding to the interior GGR points are given by*

$$\xi_i = (-1)^{i-1} \sin\left(\frac{1}{2}\cos^{-1}\tau_i\right)\sqrt{2\varpi_i}, \quad i = 1, 2, \ldots, n, \tag{4.10a}$$

$$= (-1)^{i-1} \sin\left(\cos^{-1}\tau_i\right)\sqrt{\frac{\varpi_i}{1+\tau_i}}, \quad i = 1, 2, \ldots, n. \tag{4.10b}$$

Proof. Through the change of variables $\tau = \cos\theta$ and the double-angle rule for the cosine function, we find that

$$\sqrt{1-\tau_i} = \sqrt{1-\cos\theta_i} = \sqrt{2\sin^2\left(\frac{1}{2}\theta_i\right)} = \sqrt{2}\sin\left(\frac{1}{2}\cos^{-1}\tau_i\right), \quad i = 1, 2, \ldots, n, \tag{4.11}$$

from which Eq (4.10a) is derived. Equation (4.10b) is established by realizing that

$$\sqrt{1-\tau_i} = \sqrt{\frac{1-\tau_i{}^2}{1+\tau_i}} = \frac{\sin\left(\cos^{-1}\tau_i\right)}{\sqrt{1+\tau_i}}, \quad i = 1, 2, \ldots, n, \tag{4.12}$$

which completes the proof. □

We refer to Eqs (4.10a) and (4.10b) by the trigonometric-barycentric weights. Equation (4.8b) is faster to compute and requires a smaller number of arithmetic operations compared with Eqs (4.10a) and (4.10b), but the latter two formulas may possibly produce smaller errors near $\tau = 1$, as we observed through numerical experiments. This suggests that it is better to perform Eq (4.8b) for all values of τ, except when τ is sufficiently close to 1, where we switch to the other trigonometric forms. To this end, we introduce "a switching parameter," $0 < \varepsilon \ll 1$, at which the interchange of formulas is performed. The crossover value of ε, where it becomes more accurate to use the trigonometric form, will depend on the implementation; a prescription of this strategy is outlined in Algorithms B.1 and B.2. We refer to the explicit formulas used in Algorithms B.1 and B.2 to compute the barycentric weights by the "first and second switching formulas" of the barycentric weights for the GGR points, respectively. We also denote the errors in computing the barycentric weights using Eq (4.8b), Algorithm B.1 and Algorithm B.2, by $E_{i,n}^{(\alpha)}$, for $i = 1, 2$ and 3, respectively. Figures 3 and 4 show comparisons between the three strategies, in terms of error, for certain value ranges of ε, n and α. While Algorithm B.1 does not look promising against the usual Eq (4.8b) relative to the given input data, as clearly shown in Figure 3, Figure 4 manifests that Algorithm B.2 is more numerically stable near $\varepsilon = 0.1$ and provides better approximations. We shall therefore use the latter algorithm with $\varepsilon = 0.1$ for the computation of the barycentric weights, and we refer to the rational interpolation with barycentric weights obtained through the second switching formulas as the "SR interpolation."

Remark 4.1. *The near optimal value* $\varepsilon = 0.1$ *is based on extensive numerical experiments for candidate* ε *values between* 0 *and* 1 *with no theoretical evidence.*

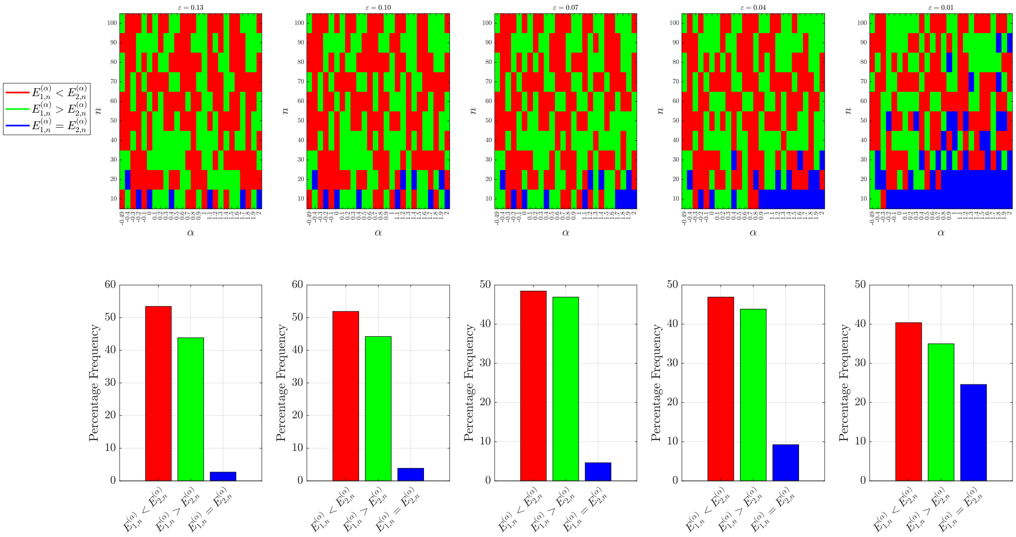

Figure 3. First row: the color maps corresponding to the switching parameter values $\varepsilon = 0.13, 0.1, 0.07, 0.04$ and 0.01. Each color map shows areas delineated by red, green and blue colors. Each color specifies whether $E^{(\alpha)}_{1,n}$ is smaller/larger/equal than/to $E^{(\alpha)}_{2,n}$ for $n = 10(10)100$ and $\alpha = -0.49, -0.4(0.1)2$. The second row shows the percentage frequency that $E^{(\alpha)}_{1,n}$ occurs as smaller/larger/equal than/to $E^{(\alpha)}_{2,n}$ in each color map. All computations were carried out using MATLAB in double-precision floating-point arithmetic.

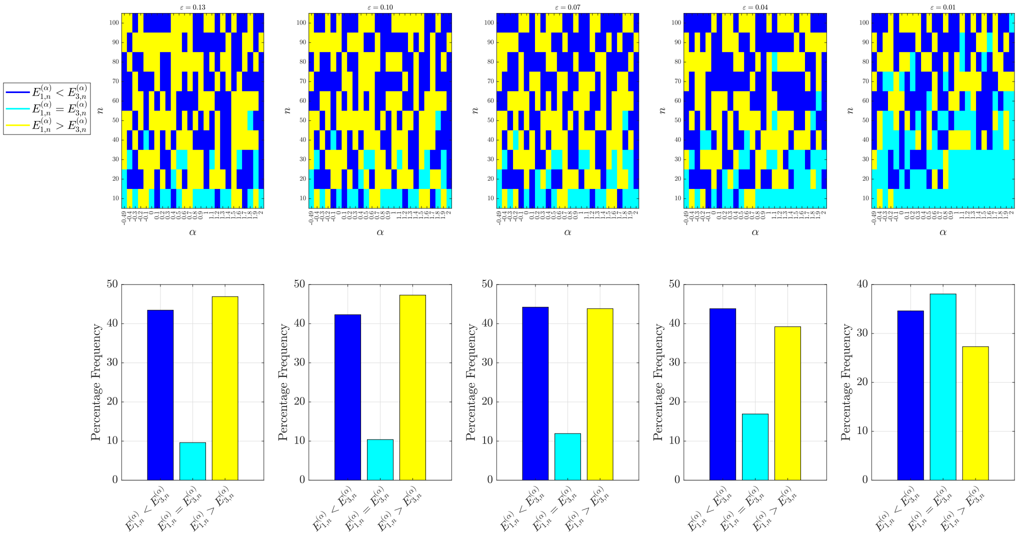

Figure 4. First row: the color maps corresponding to the switching parameter values $\varepsilon = 0.13, 0.1, 0.07, 0.04$ and 0.01. Each color map shows areas delineated by blue, cyan and yellow colors. Each color specifies whether $E^{(\alpha)}_{1,n}$ is smaller/larger/equal than/to $E^{(\alpha)}_{3,n}$ for $n = 10(10)100$ and $\alpha = -0.49, -0.4(0.1)2$. The second row shows the percentage frequency that $E^{(\alpha)}_{1,n}$ occurs as smaller/larger/equal than/to $E^{(\alpha)}_{3,n}$ in each color map. All computations were carried out using MATLAB in double-precision floating-point arithmetic.

A valuable device for measuring the quality and numerical stability of polynomial interpolations is the Lebesgue constant, as it provides a measure of how close the interpolant of a function is to the best polynomial approximant of the function. The Lebesgue constant is also very useful in assessing the quality of approximate solutions obtained through collocation (that is, the collocated solutions), as their accuracy is related to the rate at which the Lebesgue constant increases.

Let $\|f\|_{\mathbb{S}} = \sup\{|f(x)| : x \in \mathbb{S}\}$ be the uniform norm (or supremum norm) of a real-valued, bounded function f defined on a set $\mathbb{S} \subseteq \mathbb{R}$. Suppose that $y_i, i = 0, \ldots, n$ and $y_{c,n}(\tau)$ denote the exact solution values at the GGR points $\tau_i, i = 0, \ldots, n$, and the corresponding collocated solution, respectively. Let $p_n^* y$ and $p_n y$ be the best polynomial approximation to the exact solution y on $[-1, 1)$ and the Lagrange interpolating polynomial of degree at most n that interpolate the data set $\{(\tau_i, y_i)\}_{i=0}^n$, respectively. Through the uniqueness of Lagrange interpolation, one can easily show that

$$\left|p_n^* y(\tau) - p_n y(\tau)\right| = \left|\sum_{i=0}^{n} \left[p_n^* y(\tau_i) - y_i\right] \mathcal{L}_{n,i}(\tau)\right| \le \Lambda_n^{(\alpha)} \left\|y - p_n^* y\right\|_{[-1,1)}, \tag{4.13}$$

where $\Lambda_n^{(\alpha)} = \max_{-1 \le x < 1} \sum\limits_{i=0}^{n} \left|\mathcal{L}_{n,i}(\tau)\right|$ denotes the Lebesgue constant associated with GGR-based SR interpolation. Therefore,

$$\begin{aligned}\left\|y - y_{c,n}\right\|_{[-1,1)} &= \left\|y - p_n^* y + p_n^* y - p_n y + p_n y - y_{c,n}\right\|_{[-1,1)} \\ &\le \left\|y - p_n^* y\right\|_{[-1,1)} + \left\|p_n^* y - p_n y\right\|_{[-1,1)} + \left\|y_{c,n} - p_n y\right\|_{[-1,1)} \\ &\le \left(1 + \Lambda_n^{(\alpha)}\right) \left\|y - p_n^* y\right\|_{[-1,1)} + \left\|\delta y_{c,n}\right\|_{[-1,1)},\end{aligned} \tag{4.14}$$

where $\delta y_{c,n}(\tau)$ is the difference between the Lagrange interpolating polynomial and the collocated solution. When $\left\|\delta y_{c,n}\right\|_{[-1,1)} \approx 0$, $\Lambda_n^{(\alpha)}$ roughly bounds the collocation error, that is, it nearly quantifies how much larger the collocation error $\left\|y - y_{c,n}\right\|_{[-1,1)}$ is compared to the smallest possible error, $\left\|y - p_n^* y\right\|_{[-1,1)}$, in the worst case. In this case, it is obvious from Eq (4.14) that, the smaller the Lebesgue constant, the better is the predicted collocated solution in the uniform norm. In other words, the collocation error is about at most a factor $1 + \Lambda_n^{(\alpha)}$ worse than the best possible polynomial approximation. One can also clearly see that $\Lambda_n^{(\alpha)}$ depends on the location of the collocation points τ_i, $i = 0, \ldots, n$, but not on the solution values y_i, $i = 0, \ldots, n$. Since the positions of the GGR points change as α and n vary, we are interested in learning the apt choices of α that makes $\Lambda_n^{(\alpha)}$ as small as possible while holding n fixed. This can provide some useful insight into how we should select the candidate range of the α values often used for collocations based on GGR points. In [24], our findings uncovered that $\Lambda_n^{(\alpha)}$ for FGGR-based polynomial interpolation in Lagrange basis form blows up as $\alpha \to -0.5$ and monotonically increases for increasing positive values of α; see [24, Eqs (3.7) and (3.8)]. Moreover, it was noticed that $\Lambda_n^{(\alpha)}$ does not decrease monotonically for increasing negative values of α, indicating that $\Lambda_n^{(\alpha)}$ is not minimal for Chebyshev polynomials, but, rather, attains its smallest value for the Gegenbauer polynomials associated with some negative values of α; see [24, Figures 1 and 3]. It was roughly estimated in many earlier works through theoretical and numerical evidence that reasonably good α values for polynomial interpolation in basis form typically belong to the Gegenbauer collocation interval of choice, $I_{\varepsilon,r}^{G}$, defined by

$I^G_{\varepsilon,r} = \{\alpha | -1/2 + \delta \le \alpha \le r, 0 < \delta \ll 1, r \in [1,2]\}$ for reasons pertaining to the stability and accuracy of numerical schemes employing Gegenbauer polynomials as basis polynomials; see [7,20,21,24,58] and the references therein.

On the other hand, it was discovered in a number of works that Lebesgue constants for rational interpolation at equally spaced nodes are much smaller than those associated with classical polynomial interpolation [59–63]. Moreover, the Lebesgue constant of Berrut's rational interpolation [64] at equidistant nodes is smaller than the Lebesgue constant for polynomial interpolation at Chebyshev nodes [65]. Figure 5 shows the surface of the Lebesgue constant for GGR-based rational interpolation characterized by Eqs (4.4) and (4.6) with barycentric weights obtained through the second switching formulas. The surface was constructed through least-squares approximation, and it is shown together with some of its cross sections with the vertical planes $\alpha = -0.49, -0.2, -0.1, 0, 0.5, 1, 1.5$. A number of remarks deserve to be made at this point. (i) First, notice the small size of the Lebesgue constant values for GGR-based rational interpolation compared with its values for the FGGR-based polynomial interpolation in basis form; see [24, Figures 1]. (ii) $\Lambda_n^{(\alpha)}$ does not blow up as $\alpha \to -0.5$, but, rather, remains bounded. (iii) The associated Lebesgue constant grows logarithmically in the number of collocation nodes. (iv) It is also interesting to see how the Lebesgue constant drops monotonically as α increases while holding n fixed. This suggests that Legendre polynomials are generally more suited for GGR-based SR interpolations than Chebyshev polynomials. In fact, we can also observe from Figure 5 that Gegenbauer polynomials with increasing α values are associated with smaller Lebesgue constant values. This suggests that Gegenbauer polynomials with $\alpha > 1/2$ may also be more plausible to employ in SR interpolation for short/medium ranges of n values. However, the work of Elgindy and Smith-Miles [20] manifests that Gegenbauer quadratures "may become sensitive to round-off errors for positive and large values of the parameter α due to the narrowing effect of the Gegenbauer weight function," which drives the quadratures to become more extrapolatory with greater uncertainty in integral approximations; thus, the collocation is subject to a higher risk of producing meaningless results. In other words, $\left\|\delta y_{c,n}\right\|_{[-1,1)}$ may become large and the collocation error grows accordingly. It was also observed in [20] that the weight function ceases to exist near the boundaries $\tau = \pm 1$, and that its support is nonzero only on a subinterval centered at $\tau = 0$ for increasing values of $\alpha > 1$. If we refer to GGR-based collocations employing SR interpolations by the "SR collocations," then this analysis suggests that, for a relatively large collocation mesh size, SR collocations are expected to produce higher-order approximations for nonnegative α values with apparently optimal α values within/near the "Gegenbauer SR collocation interval of choice (SRCIC)" $\Upsilon^G_{\alpha_c^-,\alpha_c^+} = [\alpha_c^-, \alpha_c^+] : \alpha_c^- \approx 1/2, 1/2 \le \alpha_c^+ \le 1$; in addition, Gegenbauer polynomials with positive and large α values are generally not apt for SR interpolation/collocation. For small mesh sizes, however, there is no rule of thumb as to how should we select α, since all Lebesgue constant curves converge to the same limit as $n \to 1$. The analysis in this section requires the assumption that the problem under study is well-conditioned. For sensitive problems, the interval of choice $\Upsilon^G_{\alpha_c^-,\alpha_c^+}$ may change depending on the sources of sensitivity. In Section 5, we shall show that proper collocations of the FHOCI using any of the maps described by Eqs (3.3a) and (3.3b) entail shifting the right boundary, α_c^+, of $\Upsilon^G_{\alpha_c^-,\alpha_c^+}$ rightward as the mesh size grows large to reduce the divergence rate of the collocated solutions from the exact solutions.

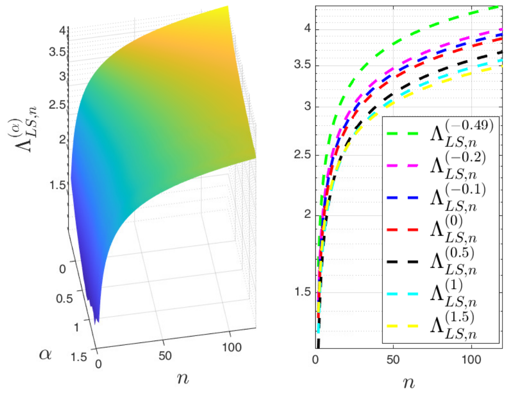

Figure 5. Left: the surface of the Lebesgue constant for GGR-based SR interpolation on the discrete rectangular domain $\{(n, \alpha) : n = 2(1)120, \alpha = [-0.499, -0.49, -0.4(0.1)1.5]\}$. The surface was constructed through least-squares approximation using curves of the form $c_1 + c_2 \ln n$ for some real parameters c_1 and c_2 with the logarithmic scale on the z axis. The right plot shows the cross sections of the surface with the vertical planes $\alpha = -0.49, -0.2, -0.1, 0, 0.5, 1, 1.5$.

From another perspective, it is interesting to mention that the Lebesgue constant is also a useful instrument in observing how the collocated solutions change as the input data are varied. By closely following the convention in [24], suppose that $\tilde{y}_i, i = 0, \ldots, n$, and $\tilde{y}_{c,n}(\tau)$ are the perturbed solution values due to round-off or input data errors and the perturbed collocated solution, respectively. Moreover, assume that $\tilde{p}_n y(\tau)$ is the Lagrange interpolating polynomial of degree at most n that interpolates the data set $\{(\tau_i, \tilde{y}_i)\}_{i=0}^{n}$. Then, we have

$$\begin{aligned}
\left\|y_{c,n} - \tilde{y}_{c,n}\right\|_{[-1,1)} &= \left\|p_n y - \delta y_{c,n} - \tilde{p}_n y + \delta \tilde{y}_{c,n}\right\|_{[-1,1)} \le \|p_n y - \tilde{p}_n y\|_{[-1,1)} + \left\|\delta y_{c,n} - \delta \tilde{y}_{c,n}\right\|_{[-1,1)} \\
&= \max_{-1 \le \tau < 1} \left| \sum_{i=0}^{n} (y_i - \tilde{y}_i)\, \mathcal{L}_{n,i}(\tau) \right| + \left\|\delta y_{c,n} - \delta \tilde{y}_{c,n}\right\|_{[-1,1)} \\
&\le \Lambda_n^{(\alpha)} \max_{0 \le i \le n} |y_i - \tilde{y}_i| + \left\|\delta y_{c,n} - \delta \tilde{y}_{c,n}\right\|_{[-1,1)}, \qquad (4.15a) \\
&\le \Lambda_n^{(\alpha)} \|y - \tilde{y}\|_{[-1,1)} + \left\|\delta y_{c,n} - \delta \tilde{y}_{c,n}\right\|_{[-1,1)}, \qquad (4.15b)
\end{aligned}$$

where $\delta \tilde{y}_{c,n}(\tau)$ is the difference between the perturbed Lagrange interpolating polynomial and the perturbed collocated solution. When $\left\|\delta y_{c,n} - \delta \tilde{y}_{c,n}\right\|_{[-1,1)}$ is relatively small, $\Lambda_n^{(\alpha)}$ nearly quantifies the larger size of the perturbation error of the collocated solution, $\left\|y_{c,n} - \tilde{y}_{c,n}\right\|_{[-1,1)}$, compared to the maximum possible perturbation error of the solution at the collocation points, $\max\limits_{0 \le i \le n} |y_i - \tilde{y}_i|$, or to the maximum solution perturbation error, $\|y - \tilde{y}\|_{[-1,1)}$, in the worst case.

4.2. Barycentric GGR-based integration matrix (GRIM) and quadratures

Consider a real-valued function f defined on the interval $[-1, 1]$ and its GGR-Based SR interpolation given by Eqs (4.4) and (4.6) and the second switching formulas of the barycentric weights. Following the work in [66], the formulas needed to construct the nonzero rows of the barycentric GRIM can be derived by integrating Eq (4.4) on the successive intervals $\left[-1, \tau_j\right], j = 1, \ldots, n$, to obtain

$$\int_{-1}^{\tau_j} P_n f(\tau)\, d\tau = \sum_{i=0}^{n} f_i \int_{-1}^{\tau_j} \mathcal{L}_{n,i}(\tau)\, d\tau, \quad j = 1, \ldots, n, \tag{4.16}$$

where $f_i = f(\tau_i)\, \forall i$. With the change of variable

$$\tau = \frac{1}{2}\left[(\tau_j + 1)t + \tau_j - 1\right], \tag{4.17}$$

we can rewrite Eq (4.16) as

$$\int_{-1}^{\tau_j} P_n f(\tau)\, d\tau = \frac{\tau_j + 1}{2} \sum_{i=0}^{n} f_i \int_{-1}^{1} \mathcal{L}_{n,i}(t; -1, \tau_j) dt, \quad j = 1, 2, \ldots, n, \tag{4.18}$$

where

$$\mathcal{L}_{n,i}\left(t; -1, \tau_j\right) = \mathcal{L}_{n,i}\left(\frac{(\tau_j + 1)t + \tau_j - 1}{2}\right), \quad \forall i, j. \tag{4.19}$$

Since the polynomials $\mathcal{L}_{n,i}\left(t; -1, \tau_j\right), i = 0, \ldots, n$, are of degree n, the integrals $\int_{-1}^{1} \mathcal{L}_{n,i}(t; -1, \tau_j) dt$ can be computed exactly using an $N = \lceil (n+1)/2 \rceil$-point LG quadrature, where $\lceil . \rceil$ denotes the ceiling function. Let $\{\bar{\tau}_k, \bar{\varpi}_k\}_{k=0}^{N}$ be the set of LG quadrature nodes and weights, respectively, where

$$\bar{\varpi}_k = \frac{2}{\left(1 - \bar{\tau}_k^2\right)\left(L'_{N+1}(\bar{\tau}_k)\right)^2}, \quad k = 0, \ldots, N, \tag{4.20}$$

and L'_{N+1} denotes the derivative of the $(N+1)$st-degree Legendre polynomial L_{N+1}. Then,

$$\int_{-1}^{1} \mathcal{L}_{n,i}(t; -1, \tau_j)\, dt = \sum_{k=0}^{N} \bar{\varpi}_k \mathcal{L}_{n,i}(\bar{\tau}_k; -1, \tau_j). \tag{4.21}$$

Hence, Eqs (4.18) and (4.21) yield the GGR-SR quadrature rule

$$\int_{-1}^{\tau_j} f(\tau)\, d\tau \approx \sum_{i=0}^{n} q_{j,i} f_i = \mathbf{Q}\boldsymbol{f}, \quad j = 0, \ldots, n, \tag{4.22}$$

where $\boldsymbol{f} = [f_0, f_1, \ldots, f_n]^\top$ and $q_{j,i}, i, j = 0, \ldots, n$, are the elements of the first-order barycentric GRIM $\mathbf{Q}$ given by

$$q_{j,i} = \begin{cases} 0, \quad j = 0,\ i = 0, \ldots, n, \\ \dfrac{\tau_j + 1}{2} \displaystyle\sum_{k=0}^{N} \bar{\varpi}_k \mathcal{L}_{n,i}(\bar{\tau}_k; -1, \tau_j), \quad j = 1, 2, \ldots, n,\ i = 0, \ldots, n. \end{cases} \tag{4.23}$$

We denote the jth row of $\mathbf{Q}$ by $\mathbf{Q}_j\, \forall j$. The derivation of the formulas required to construct the GGR-based differentiation matrix (GRDM) in barycentric form is described in Appendix A.

4.3. IPS rational collocation of the FHOC at the GGR points

Let $\boldsymbol{\tau}_n = [\tau_0, \tau_1, \ldots, \tau_n]^\top, \tilde{\boldsymbol{x}}_i = \tilde{\boldsymbol{x}}(\tau_i), \tilde{\boldsymbol{u}}_i = \tilde{\boldsymbol{u}}(\tau_i), \boldsymbol{f}_i = \boldsymbol{f}(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{u}}_i), f_{i,j} = f_i(\tilde{\boldsymbol{x}}_j, \tilde{\boldsymbol{u}}_j)\ \forall i, j$, and

$$\mathcal{J}_n = \left[\int_{-1}^{\tau_0} T'(\tau)\boldsymbol{f}(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau))\, d\tau, \ldots, \int_{-1}^{\tau_n} T'(\tau)\boldsymbol{f}(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau))\, d\tau\right]. \tag{4.24}$$

Then, collocating Eq (3.1d) at the GGR nodes yields

$$\tilde{\boldsymbol{x}}(\boldsymbol{\tau}_n) = \operatorname{vec}(\mathcal{J}_n) + \boldsymbol{x}_0 \otimes \boldsymbol{1}_{n+1} \approx \hat{\mathbf{M}} + \boldsymbol{x}_0 \otimes \boldsymbol{1}_{n+1}, \tag{4.25}$$

where

$$\hat{\mathbf{M}} = \operatorname{vec}(\mathbf{M}) : \quad \mathbf{M} = \mathbf{Q}\left(\mathbf{F} \circ \left[T'(\boldsymbol{\tau}_n) \otimes \boldsymbol{1}_{n_x}^\top\right]\right), \tag{4.26}$$

$\mathbf{F} = [\mathbf{F}_{1,n}, \ldots, \mathbf{F}_{n_x,n}], \mathbf{F}_{i,n} = [f_{i,0}, \ldots, f_{i,n}]^\top\ \forall i, \boldsymbol{1}_n$ is the all-ones column vector of size n and $[.,.]$, "vec", $\circ$ and $\otimes$ denote the horizontal matrix concatenation, the vectorization of a matrix, the Hadamard product, and the Kronecker product, respectively. Let $\tau_{n+1} = 1$ and define $\mathbf{Q}_{n+1} = (q_{n+1,i})_{0\le i\le n} : q_{n+1,i} = \sum_{k=0}^{N} \bar{\varpi}_k \mathcal{L}_{n,i}(\bar{\tau}_k)\ \forall i$; then, the discrete cost functional J can be approximated numerically using the LG quadrature, as follows:

$$J \approx J_n = \mathbf{Q}_{n+1}(T'(\boldsymbol{\tau}_n) \circ \boldsymbol{g}), \tag{4.27}$$

where $\boldsymbol{g} = [g_0, \ldots, g_n]^\top : g_i = g(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{u}}_i)\ \forall i$. Hence, the FHOCI (3.1a), (3.1c) and (3.1d) can now be converted into a NLP in which the goal is to minimize the discrete cost functional given by Eq (4.27), as subject to the nonlinear system of equations given by Eq (4.25). If we define the image of the collocation points set $\mathbb{S}_n$ under the transformation T by $\mathbb{S}_n^T = \{t_k : t_k = T(\tau_k),\ k = 0, \ldots, n\}$, and denote $\boldsymbol{x}(t_i)$ and $\boldsymbol{u}(t_i)$ by $\boldsymbol{x}_i$ and $\boldsymbol{u}_i\ \forall i$, respectively, then the NLP can be solved using well-developed optimization software for the unknowns $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i$, $i = 1, \ldots, n$, and $\tilde{\boldsymbol{u}}_j = \boldsymbol{u}_j$, $j = 0, \ldots, n$. The approximate optimal state and control variables can then be calculated at any point $t \in [0, \infty)$ through the PS expansions

$$\boldsymbol{x}(t) = \boldsymbol{x}(\boldsymbol{t}_n^\top)\,\mathcal{L}_n(t), \tag{4.28a}$$

$$\boldsymbol{u}(t) = \boldsymbol{u}(\boldsymbol{t}_n^\top)\,\mathcal{L}_n(t), \tag{4.28b}$$

where $\boldsymbol{t}_n = [t_0, t_1, \ldots, t_n]^\top$ and $\mathcal{L}_n(t) = [\mathcal{L}_{n,0}(t), \mathcal{L}_{n,1}(t), \ldots, \mathcal{L}_{n,n}(t)]^\top$. In the special case, when $T = T_{1,L}^{(\alpha)}$, one can easily show that the NLP can be written as follows:

$$\min J_n = 2L\mathbf{Q}_{n+1}[\boldsymbol{g} \oslash (\boldsymbol{1}_{n+1} - \boldsymbol{\tau}_n)_2], \tag{4.29a}$$

subject to

$$\tilde{\boldsymbol{x}}(\boldsymbol{\tau}_n) \approx 2L\hat{\mathbf{M}}_1 + \boldsymbol{x}_0 \otimes \boldsymbol{1}_{n+1}, \tag{4.29b}$$

where $\oslash$ denotes the Hadamard division, $(\boldsymbol{v})_r = \underbrace{\boldsymbol{v} \circ \boldsymbol{v} \circ \ldots \circ \boldsymbol{v}}_{r \text{ times}}$, for any vector $\boldsymbol{v}$, and

$$\hat{\mathbf{M}}_1 = \operatorname{vec}(\mathbf{M}_1) : \quad \mathbf{M}_1 = \mathbf{Q}\left(\mathbf{F} \oslash \left[(\boldsymbol{1}_{n+1} - \boldsymbol{\tau}_n)_2 \otimes \boldsymbol{1}_{n_x}^\top\right]\right). \tag{4.30}$$

Furthermore, when $T = T_{2,L}^{(\alpha)}$, the NLP can be formulated as follows:

$$\min J_n = L\mathbf{Q}_{n+1}[\boldsymbol{g} \oslash (\boldsymbol{1}_{n+1} - \boldsymbol{\tau}_n)], \tag{4.31a}$$

subject to

$$\tilde{\boldsymbol{x}}\left(\boldsymbol{\tau}_n\right) \approx L\hat{\mathbf{M}}_2 + \boldsymbol{x}_0 \otimes \boldsymbol{1}_{n+1}, \tag{4.31b}$$

where

$$\hat{\mathbf{M}}_2 = \operatorname{vec}\left(\mathbf{M}_2\right): \quad \mathbf{M}_2 = \mathbf{Q}\left(\mathbf{F} \oslash \left[\left(\boldsymbol{1}_{n+1} - \boldsymbol{\tau}_n\right) \otimes \boldsymbol{1}_{n_x}^{\top}\right]\right). \tag{4.32}$$

We refer to the NLPs described by Eqs (4.29a), (4.29b), (4.31a) and (4.31b) by NLP1 and NLP2, respectively. We also refer to the present collocation method as the "GGR-IPS" method; the acronyms "GGR-IPS1" and "GGR-IPS2" stand for the GGR-IPS method performed using the parametric maps $T_{1,L}^{(\alpha)}$ and $T_{2,L}^{(\alpha)}$, respectively, while "GGR-IPS12" stands for the GGR-IPS method performed using either maps $T_{1,L}^{(\alpha)}$ and $T_{2,L}^{(\alpha)}$. Notice that the Lagrangian associated with the NLP described by Eqs (4.25) and (4.27) is defined by

$$\mathfrak{L} = \mathbf{Q}_{n+1}\left(\boldsymbol{T}_n' \circ \boldsymbol{g}\right) + \boldsymbol{r}^{\top}\left(\hat{\mathbf{M}} + \boldsymbol{x}_0 \otimes \boldsymbol{1}_{n+1} - \tilde{\boldsymbol{x}}\left(\boldsymbol{\tau}_n\right)\right), \tag{4.33}$$

where $\boldsymbol{T}_n' = T'\left(\boldsymbol{\tau}_n\right)$ and $\boldsymbol{r} = \left[r_{10}, \ldots, r_{1n}, \ldots, r_{n_x0}, \ldots, r_{n_xn}\right]^{\top}$ is the vector of Lagrange multipliers. Therefore, the Karush–Kuhn–Tucker necessary conditions of optimality are given by

$$\underset{\tilde{\boldsymbol{x}}}{\nabla}\mathfrak{L} = \mathbf{Q}_{n+1}\left[\left(\boldsymbol{T}_n' \otimes \boldsymbol{1}_{n_xn}^{\top}\right) \circ \underset{\tilde{\boldsymbol{x}}}{\nabla}\boldsymbol{g}\right] + \boldsymbol{r}^{\top}\left[\left(\mathbf{I}_{n_x} \otimes \left[\mathbf{Q} \circ \left(\boldsymbol{T}_n'^{\top} \otimes \boldsymbol{1}_{n+1}\right)\right]\right)\underset{\tilde{\boldsymbol{x}}}{\nabla}\hat{\mathbf{F}} - \mathbf{I}_{n_x} \otimes \mathbf{E}\right] = \boldsymbol{0}, \tag{4.34}$$

$$\underset{\tilde{\boldsymbol{u}}}{\nabla}\mathfrak{L} = \mathbf{Q}_{n+1}\left[\left(\boldsymbol{T}_n' \otimes \boldsymbol{1}_{n_u(n+1)}^{\top}\right) \circ \underset{\tilde{\boldsymbol{u}}}{\nabla}\boldsymbol{g}\right] + \boldsymbol{r}^{\top}\left[\left(\mathbf{I}_{n_x} \otimes \left[\mathbf{Q} \circ \left(\boldsymbol{T}_n'^{\top} \otimes \boldsymbol{1}_{n+1}\right)\right]\right)\underset{\tilde{\boldsymbol{u}}}{\nabla}\hat{\mathbf{F}}\right] = \boldsymbol{0}, \tag{4.35}$$

where the operators $\underset{\tilde{\boldsymbol{x}}}{\nabla} = \left[\frac{\partial}{\partial\tilde{x}_{11}} \cdots \frac{\partial}{\partial\tilde{x}_{1n}} \cdots \frac{\partial}{\partial\tilde{x}_{n_x1}} \cdots \frac{\partial}{\partial\tilde{x}_{n_xn}}\right], \underset{\tilde{\boldsymbol{u}}}{\nabla} = \left[\frac{\partial}{\partial\tilde{u}_{10}} \cdots \frac{\partial}{\partial\tilde{u}_{1n}} \cdots \frac{\partial}{\partial\tilde{u}_{n_u0}} \cdots \frac{\partial}{\partial\tilde{u}_{n_un}}\right]$, $\mathbf{I}_n$ is the identity matrix of size $n, \hat{\mathbf{F}} = \operatorname{vec}\left(\mathbf{F}\right), \mathbf{E} = \left[\boldsymbol{0}_n^{\top}; \mathbf{I}_n\right]$ and $[.;.]$ denotes the vertical matrix concatenation.

5. Error and convergence analyses

In this section, we derive the truncation error bounds for Eqs (4.25) and (4.27) and their convergence rates.

Theorem 5.1. *Let $f \in C^{n+1}[-1,1)$ be approximated by a Gegenbauer interpolant P_nf based upon the GGR points set $\mathbb{S}_n$, as defined by Eq* (4.4)*. Then, there exist some numbers $\xi_i \in (-1,1), i = 0, \ldots, n$, such that the truncation error of the GGR-SR quadrature rule of Eq* (4.22) *is given by*

$$_fE_n\left(\tau_i, \xi_i\right) = \frac{f^{(n+1)}\left(\xi_i\right)}{(n+1)!K_{n+1}^{(\alpha)}}\int_{-1}^{\tau_i}\mathcal{G}_{n+1}^{(\alpha)}(\tau)\,d\tau \quad \forall i, \tag{5.1}$$

where $K_n^{(\alpha)} = 2^{n-1}\frac{\Gamma(n+\alpha)\Gamma(2\alpha+1)}{\Gamma(n+2\alpha)\Gamma(\alpha+1)}, n = 0, 1, \ldots.$

Proof. By definition, we can write

$$f(\tau) = \sum_{k=0}^{n} f_k\,\mathcal{L}_{n,k}(\tau) + {}_fE_n\left(\tau, \xi\right), \quad \forall\tau \in [-1,1), \tag{5.2}$$

for some $\xi \in (-1,1)$, where ${}_fE_n$ is the interpolation truncation error at the GGR points given by

$$ {}_fE_n(\tau,\xi) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{n} (\tau - \tau_k)\,. \tag{5.3}$$

The proof is established by realizing that $\mathcal{G}_{n+1}^{(\alpha)}(\tau) = K_{n+1}^{(\alpha)} \prod_{k=0}^{n} (\tau - \tau_k)$ and integrating Eq (5.2) on $[-1,\tau_i)\ \forall i$. □

The following result is a direct corollary of Theorem 5.1 by letting $\eta : [-1,1) \to \mathbb{R}$ and $\psi_k : [-1,1) \to \mathbb{R}$ such that $\eta(\tau) = T'(\tau) g(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau))$ and $\psi_k(\tau) = T'(\tau) f_k(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau))$ for each $k = 1,\ldots,n_x$.

Corollary 5.1. *The truncation errors of Eq* (4.27) *and each approximate equation of System* (4.25)*,*

$$\tilde{x}_k(\tau_j) = \int_{-1}^{\tau_j} \psi_k(\tau)\, d\tau + x_{k,0}, \tag{5.4}$$

of the Integral Constraints System (3.1d) *at each point $\tau_j \in \mathbb{S}_n$ are given by*

$$ {}_\eta E_n(\zeta) = \frac{\eta^{(n+1)}(\zeta)}{(n+1)!K_{n+1}} \int_{-1}^{1} \mathcal{G}_{n+1}^{(\alpha)}(\tau) d\tau \tag{5.5}$$

and

$$ {}_{\psi_k}E_n\left(\tau_j,\xi_j\right) = \frac{\psi_k^{(n+1)}\left(\xi_j\right)}{(n+1)!K_{n+1}} \int_{-1}^{\tau_j} \mathcal{G}_{n+1}^{(\alpha)}(\tau)\, d\tau, \quad k = 1,\ldots,n_x, \quad j = 0,\ldots,n, \tag{5.6}$$

respectively, where $\zeta, \xi_j \in (-1,1)\ \forall j$.

The following upper bounds on the truncation errors of Eqs (4.25) and (4.27) can be deduced from [24, Theorem 5.1].

Theorem 5.2. *Let $\psi_k \in C^{n+1}[-1,1)$ and $\left\|\psi_k^{(n+1)}\right\|_{[-1,1)} = A_{\psi_k,n} \in \mathbb{R}^+\ \forall k$ for some constant $A_{\psi_k,n}$ dependent on n and k. Then, there exist some positive constants $B_1^{(\alpha)}$ and $C_1^{(\alpha)}$ dependent on α and independent of n such that the truncation errors of System* (4.25) *at each point $\tau_j \in \mathbb{S}_n$ are bounded by the following inequalities:*

$$\left|{}_{\psi_k}E_n\left(\tau_j,\xi_j\right)\right| \le \frac{A_{\psi_k,n}\Gamma(n+2\alpha+1)\Gamma(\alpha+1)\left(\tau_j+1\right)}{2^n(n+1)!\Gamma(n+\alpha+1)\Gamma(2\alpha+1)} \left\|\mathcal{G}_{n+1}^{(\alpha)}\right\|_{[-1,1)}, \quad k = 1,\ldots,n_x, \quad j = 0,\ldots,n, \tag{5.7}$$

where $\xi_j \in (-1,1)\ \forall j$ and

$$\left\|\mathcal{G}_{n+1}^{(\alpha)}\right\|_{[-1,1)} = \begin{cases} 2, \quad n \ge 0,\ \alpha \ge 0, \\ \dfrac{\Gamma\left(\alpha+\frac{1}{2}\right)\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{\pi}\Gamma\left(\alpha+\frac{n+1}{2}\right)}\left(1+\sqrt{\dfrac{n+1}{2\alpha+n+1}}\right), & \dfrac{n}{2} \in \mathbb{Z}_0^+ \wedge \dfrac{-1}{2} < \alpha < 0, \\ \dfrac{\Gamma\left(\alpha+\frac{1}{2}\right)\left(\sqrt{n\,(2\alpha+n)}+n\right)\Gamma\left(\frac{n}{2}\right)}{2\sqrt{\pi}\,\Gamma\left(\frac{n}{2}+\alpha+1\right)}, & \dfrac{n+1}{2} \in \mathbb{Z}^+ \wedge \dfrac{-1}{2} < \alpha < 0. \end{cases} \tag{5.8}$$

Moreover,

$$\left|{}_{\psi_k}E_n\left(\tau_j,\xi_j\right)\right| \le B_1^{(\alpha)}\left(\frac{e}{2}\right)^n \frac{1+\tau_j}{n^{n+\frac{3}{2}-\alpha}}, \quad \forall\alpha \ge 0, \tag{5.9}$$

and

$$\left|{}_{\psi_k}E_n\left(\tau_j,\xi_j\right)\right| \underset{\sim}{<} C_1^{(\alpha)}\left(\frac{e}{2}\right)^n \frac{1+\tau_j}{n^{n+\frac{3}{2}}},\quad \forall -1/2<\alpha<0, \tag{5.10}$$

as $n \to \infty$, where $\underset{\sim}{<}$ means "less than or asymptotically equal to."

Theorem 5.3. *Let $\eta \in C^{n+1}[-1,1)$ and $\left\|\eta^{(n+1)}\right\|_{[-1,1)} = A_{\eta,n} \in \mathbb{R}^+$ for some constant $A_{\eta,n}$ dependent on n. Then, there exist some positive constants $B_2^{(\alpha)}$ and $C_2^{(\alpha)}$ dependent on α and independent of n such that the truncation error of Eq* (4.27) *is bounded by the following inequality:*

$$\left|{}_{\eta}E_n(\zeta)\right| \le \frac{A_{\eta,n}\Gamma(n+2\alpha+1)\Gamma(\alpha+1)}{2^{n-1}(n+1)!\Gamma(n+\alpha+1)\Gamma(2\alpha+1)}\left\|\mathcal{G}_{n+1}^{(\alpha)}\right\|_{[-1,1)}, \tag{5.11}$$

where $\zeta \in (-1,1)$. Moreover,

$$\left|{}_{\eta}E_n(\zeta)\right| \le B_2^{(\alpha)}\left(\frac{e}{2}\right)^n \frac{1}{n^{n+\frac{3}{2}-\alpha}},\quad \forall \alpha \ge 0, \tag{5.12}$$

and

$$\left|{}_{\eta}E_n(\zeta)\right| \underset{\sim}{<} C_2^{(\alpha)}\left(\frac{e}{2}\right)^n \frac{1}{n^{n+\frac{3}{2}}},\quad \forall -1/2<\alpha<0, \tag{5.13}$$

as $n \to \infty$.

Divergence of typical IPS collocation schemes of the FHOCI at any large mesh grid of Gauss type when $T = T_{1,L}^{(\alpha)}$ or $T = T_{2,L}^{(\alpha)}$

In this section, we derive some striking results regarding the convergence of typical collocation schemes of the FHOCI described by Eqs (3.1a), (3.1c) and (3.1d) when $T \in \{T_{1,L}^{(\alpha)}, T_{2,L}^{(\alpha)}\}$ and the mesh grid is large and of Gauss type. While the proof pertains to the FHOCI and employs the GGR points as the collocation points, it can be generalized to the usual form of the FHOC described by Eqs (3.1a)–(3.1c) and any large collocation points of Gauss type, causing it to become of considerably greater interest. We derive these interesting divergence results in the following two corollaries.

Corollary 5.2. *Let $T \in \{T_{1,L}^{(\alpha)}, T_{2,L}^{(\alpha)}\}$, and suppose that $\exists\, \hat{k} \in \{1,\ldots,n_x\} : \psi_{\hat{k}} \in C^n[-1,1)$, $0 < \left\|f_{\hat{k}}\right\|_{[-1,1)} < \infty$ and $0 \le \left\|\frac{d^j}{d\tau^j}f_{\hat{k}}\right\|_{[-1,1)} < \infty \quad \forall j = 1,\ldots,n+1$; then, the upper truncation error bounds of the Approximated System* (4.25) *diverge at each collocation point as $n \to \infty$ for any map scaling parameter value L.*

Proof. By the general Leibniz rule, the $(n+1)$st derivative of $\psi_{\hat{k}}$ is given by

$$\psi_{\hat{k}}^{(n+1)}(\tau) = \sum_{j=0}^{n+1}\binom{n+1}{j}T^{(n+2-j)}(\tau)\frac{d^j}{d\tau^j}f_{\hat{k}}\left(\tilde{\boldsymbol{x}}(\tau),\tilde{\boldsymbol{u}}(\tau)\right), \tag{5.14}$$

with

$$\left\|\psi_{\hat{k}}^{(n+1)}\right\|_{[-1,1)} = \sum_{j=0}^{n+1}\binom{n+1}{j}\left\|T^{(n+2-j)}\right\|_{[-1,1)}\left\|\frac{d^j}{d\tau^j}f_{\hat{k}}\right\|_{[-1,1)}. \tag{5.15}$$

Let $T = T^{(\alpha)}_{1,L}$, and notice that $\left(T^{(\alpha)}_{1,L}\right)^{(m)}(\tau) = \dfrac{2L(m)!}{(1-\tau)^{m+1}}\ \forall m \in \mathbb{Z}^+$, which is a monotonically increasing function for increasing values of τ as can be clearly seen in Figure 6. Therefore, $A_{\psi_{\hat{k}},n} = O\left(\left\|\left(T^{(\alpha)}_{1,L}\right)^{(n+2)}\right\|_{[-1,1)}\right)$. From Theorem 5.2, there exist some positive constants $\hat{B}^{(\alpha)}_1$ and $\hat{C}^{(\alpha)}_1$ dependent on α and independent on n such that

$$\left|{}_{\psi_{\hat{k}}}E_n\left(\tau_j, \xi_j\right)\right| \le L\hat{B}^{(\alpha)}_1 (n+2)! \left(\frac{e}{2}\right)^n \frac{1+\tau_j}{n^{n+\frac{3}{2}-\alpha}} \left\|(1-\tau)^{-n-3}\right\|_{[-1,1)}, \quad \forall \alpha \ge 0, \tag{5.16a}$$

and

$$\left|{}_{\psi_{\hat{k}}}E_n\left(\tau_j, \xi_j\right)\right| \lesssim L\hat{C}^{(\alpha)}_1 (n+2)! \left(\frac{e}{2}\right)^n \frac{1+\tau_j}{n^{n+\frac{3}{2}}} \left\|(1-\tau)^{-n-3}\right\|_{[-1,1)}, \quad \forall -1/2 < \alpha < 0, \tag{5.16b}$$

from which we realize that the upper bound of $\left|{}_{\psi_{\hat{k}}}E_n\right|$ at each collocation point τ_j diverges as $n \to \infty$. Consider now the case when $T = T^{(\alpha)}_{2,L}$. By a similar argument, notice first that $\left(T^{(\alpha)}_{2,L}\right)^{(m)}(\tau) = \dfrac{L(m-1)!}{(1-\tau)^m}\ \forall m \in \mathbb{Z}^+$ is also a monotonically increasing function for increasing values of τ, as shown in Figure 6. Therefore, $A_{\psi_{\hat{k}},n} = O\left(\left\|\left(T^{(\alpha)}_{2,L}\right)^{(n+2)}\right\|_{[-1,1)}\right)$. From Theorem 5.2, there exist some positive constant $\hat{B}^{(\alpha)}_2$ and $\hat{C}^{(\alpha)}_2$ dependent on α and independent on n such that

$$\left|{}_{\psi_{\hat{k}}}E_n\left(\tau_j, \xi_j\right)\right| \le L\hat{B}^{(\alpha)}_2 (n+1)! \left(\frac{e}{2}\right)^n \frac{1+\tau_j}{n^{n+\frac{3}{2}-\alpha}} \left\|(1-\tau)^{-n-2}\right\|_{[-1,1)}, \quad \forall \alpha \ge 0, \tag{5.17a}$$

and

$$\left|{}_{\psi_{\hat{k}}}E_n\left(\tau_j, \xi_j\right)\right| \lesssim L\hat{C}^{(\alpha)}_2 (n+1)! \left(\frac{e}{2}\right)^n \frac{1+\tau_j}{n^{n+\frac{3}{2}}} \left\|(1-\tau)^{-n-2}\right\|_{[-1,1)}, \quad \forall -1/2 < \alpha < 0, \tag{5.17b}$$

from which we observe that the upper bound of $\left|{}_{\psi_{\hat{k}}}E_n\right|$ at each collocation point τ_j diverges as $n \to \infty$. □

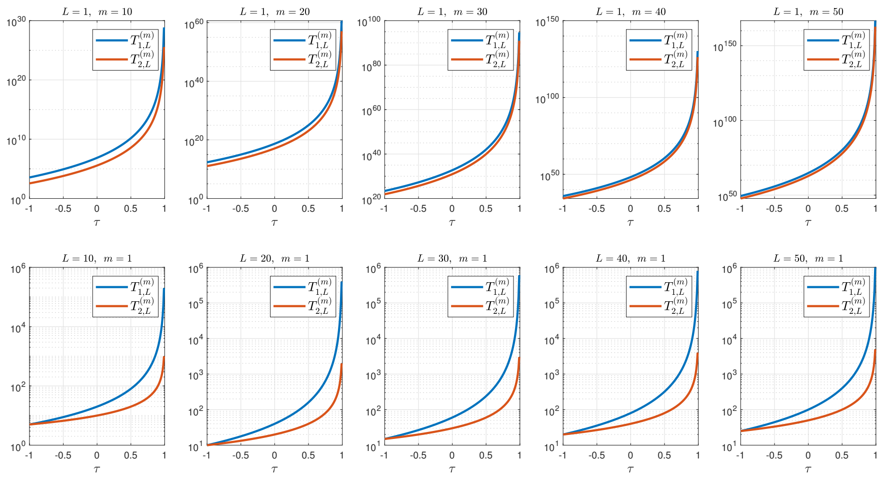

Figure 6. mth-order derivatives of $T^{(\alpha)}_{1,L}$ and $T^{(\alpha)}_{2,L}$ versus τ in log-lin scale for several values of L and m. The superscript of $T^{(\alpha)}_{i,L}, i = 1, 2$ has been omitted in the plots.

Theorem 5.2 and Corollary 5.2 reveal an interesting fact. Under their assumptions, the proposed method is expected to converge with an exponential rate to near-optimal solutions for increasing n values within a relatively small n value range, as indicated by inequality (5.7), but, as n grows large, the constant $A_{\psi_{\hat{k}},n}$ grows exponentially fast and ultimately dominates the error bounds when $n \to \infty$, as implied by the inequalities (5.16a)–(5.17b), regardless of how well we choose the map scaling parameter value L. In fact, the asymptotic results of Corollary 5.2 manifest that, for increasing large n values, reducing the L value abates the divergence of the approximations at the outset, but, as n grows larger, this approach fails to cope with the soaring values of n powers of the factors $1/(1-\tau_j)$ at the mesh points τ_j for sufficiently close mesh points values to $\tau = 1$; thus, divergence is inevitable.

While the above forward error analysis may be too pessimistic and may reject solutions that are sufficiently accurate, another concern arises when we analyze the sensitivity of NLP1 and NLP2 associated with the maps $T_{1,L}^{(\alpha)}$ and $T_{2,L}^{(\alpha)}$ to input data errors. Observe that both problems require the computations of the maps $T'_{1,L}$ and $T'_{2,L}$, which are ill-conditioned for arguments near 1. In particular, suppose that $\tau \approx 1$ with a small perturbation h to τ. Then, the absolute errors in computing $T'_{1,L}(\tau)$ and $T'_{2,L}(\tau)$ are given by

$$\left|T'_{1,L}(\tau+h) - T'_{1,L}(\tau)\right| \approx \frac{4L\,|h|}{(1-\tau)^3}, \tag{5.18a}$$

$$\left|T'_{2,L}(\tau+h) - T'_{2,L}(\tau)\right| \approx \frac{L\,|h|}{(1-\tau)^2}; \tag{5.18b}$$

hence, the relative errors are $\dfrac{2|h|}{1-\tau}$ and $\dfrac{|h|}{1-\tau}$, respectively, which blow up as $\tau \to 1$. Recall that GGR points cluster near ± 1 as $n \to \infty$, so the sensitivity of the problem of calculating the maps' derivative functions $T'_{1,L}$ and $T'_{2,L}$ at arguments near 1 increases for increasing values of n. For example, let $\tau = 0.9999999999999$ be an exact argument value and consider its approximation $\hat{\tau} = 0.9999999999998$ with a small perturbation of about 9.99×10^{-14} to τ. Then, the relative error in the input value is about $10^{-11}\%$. However, the relative errors in computing $T'_{1,L}(\tau)$ and $T'_{2,L}(\tau)$ are nearly 75% and 50%. Hence, the relative changes in evaluating $T'_{1,L}(\tau)$ and $T'_{2,L}(\tau)$ are about 7.5 and 5 trillion times larger than the relative change in the input value in respective order. This example shows that increasing the mesh size shifts the positive collocation points closer and closer toward $\tau = 1$, and that wild ill-conditioning ultimately occurs, as the sensitivity of NLP2 progressively stiffens for arguments near 1. Therefore, one should keep in mind that reducing L may still improve the approximations for a certain range of n values; nonetheless, this strategy is not prone to producing accurate approximations for relatively large values of n, in general, since both NLP1 and NLP2 are ill-conditioned near $\tau = 1$. It is noteworthy to mention here that this sensitivity of NLP1 and NLP2 near $\tau = 1$ is foreseen to relax or disappear if g and $f_k\ \forall k$ decay exponentially fast such that $\lim_{\tau\to 1} T'(\tau) g\left(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau)\right) = 0$ and $\lim_{\tau\to 1} T'(\tau) \boldsymbol{f}(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau)) = \boldsymbol{0}$. Under a similar proof to that of Corollary 5.2, one can derive the following second divergence result.

Corollary 5.3. *Let $T \in \{T_{1,L}^{(\alpha)}, T_{2,L}^{(\alpha)}\}$, $\eta \in C^n[-1,1)$, $0 < \|g\|_{[-1,1)} < \infty$ and $0 \le \left\| \dfrac{d^j}{d\tau^j} g \right\|_{[-1,1)} < \infty\ \forall j = 1, \ldots, n+1$; then, the upper truncation error bound of Eq (4.27) diverges as $n \to \infty$ for any map scaling parameter value L.*

The present analysis begs another interesting question: Which map should we use if we desire to implement the proposed method? For small/medium ranges of n values, the answer is a bit elusive; however, for large values, it seems that we have a crystal clear answer, as shown by the following corollary.

Corollary 5.4. *A Gegenbauer-Gauss collocation of the FHOCI using the map $T_{1,L}^{(\alpha)}$ generally diverges faster than that achieved by applying the method lumped with the map $T_{2,L}^{(\alpha)}$ when $n \to \infty$.*

Proof. The faster divergence exhibited using the map $T_{1,L}^{(\alpha)}$ compared with $T_{2,L}^{(\alpha)}$ as $n \to \infty$ can be easily justified by using the inequalities (5.16a)–(5.17b). Moreover, since $T'_{1,L}(\tau)$ grows faster than $T'_{2,L}(\tau)$ by a factor of $2/(1-\tau)$, which blows up as $\tau \to 1$, the ill-conditioning of $T'_{1,L}$ is clearly more severe than that of $T'_{2,L}$ for values of $\tau \approx 1$. □

Corollary 5.4 manifests that the map $T_{2,L}^{(\alpha)}$ is more likely a better choice than $T_{1,L}^{(\alpha)}$ for large n values. We end this section by drawing the attention of the reader to the fact that integral reformulations of various mathematical models have received considerable attention in the literature because they often produce well-conditioned linear systems. While numerical quadratures and integration matrices are generally more stable than numerical differentiation operators and matrices, there is no strong reason to expect that standard PS collocations of the FHOC in its strong differential form, as obtained by using a single mesh grid of Gauss type and maps like $T_{1,L}^{(\alpha)}$ and $T_{2,L}^{(\alpha)}$, would exhibit any merits over the current method, and they would ultimately diverge for a large mesh grid size. These considerations lead naturally to the following interesting conjecture.

Conjecture 5.1. *Classical Jacobi polynomial collocations of the FHOC in differential/integral form obtained through maps like $T_{1,L}^{(\alpha)}$ and $T_{2,L}^{(\alpha)}$ will likely diverge as the mesh size grows large if the computations are carried out using floating-point arithmetic and the discretizations use a single mesh grid, regardless of whether they are of Gauss/GR type or equally spaced. The former divergence case is a direct result of the present divergence analysis, while the latter case is due to Runge's phenomenon and the ill-conditioning of polynomial interpolation at equally spaced nodes as the degree of the polynomial grows.*

6. Numerical experiments

This section presents the results of some numerical experiments on three test examples, which demonstrate the accuracy and efficiency of the proposed GGR-IPS12 methods for small-medium-range mesh grid sizes and verifies the inevitable divergence as the mesh size grows large. All numerical experiments were carried out using MATLAB R2022a software installed on a personal laptop equipped with a 2.9 GHz AMD Ryzen 7 4800H CPU and 16 GB memory running on a 64-bit Windows 11 operating system. The NLPs obtained through the GGR-IPS12 methods were solved using either (i) MATLAB fmincon solver with the interior-point algorithm (fmincon-int) and sqp algorithm (fmincon-sqp), or (ii) the augmented Lagrange multiplier method [67, 68] integrated with a modified Broyden–Fletcher–Goldfarb–Shanno method and a Chebyshev PS line search method [69], which is henceforth referred to as the "EALMM." It should be clearly understood by the reader when we coin the name of the current collocation method with any NLP solver that we are implementing them both to solve the FHOCI. For example, the acronym GGR-IPS12-EALMM stands for the GGR-IPS12 methods combined with the EALMM. In all numerical tests, the exact optimal state and control variables were calculated using MATLAB with 30 digits of precision maintained in internal computations. The fmincon solver was carried out using the stopping criteria TolFun = TolX = $10^{-12}, 10^{-15}, 10^{-15}$ for Examples 1–3, respectively. The augmented Lagrange multiplier method was terminated when the lower bound on the change in the augmented Lagrangian function value during a step does not exceed 10^{-12}. All experiments were conducted using the parameters values $L \in \{0.25(0.25)10\}$ and $\alpha \in \{-0.4(0.1)2\}$. Most of the numerical simulations were performed using two sets of initial guesses: $\boldsymbol{\Omega}_1 = \{(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) : \tilde{\boldsymbol{x}}_0 = \boldsymbol{1}_{n_x}, \tilde{\boldsymbol{u}}_0 = \boldsymbol{1}_{n_u}\}$ and $\boldsymbol{\Omega}_2 = \{(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) : \tilde{\boldsymbol{x}}_0 = 0.5\boldsymbol{1}_{n_x}, \tilde{\boldsymbol{u}}_0 = 0.5\boldsymbol{1}_{n_u}\}$; henceforth, $\boldsymbol{\Omega}_1 \cup \boldsymbol{\Omega}_2$ is denoted by $\boldsymbol{\Omega}$. Furthermore, by AE_J and $MAE_{x,u}$, we mean the absolute error in the objective function value and the maximum absolute error of the state and control variables in respective order.

Example 1. Consider the IHOC (2.1)–(2.3) with $g(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \left(\ln^2 x(t) + u^2(t)\right)/2, f(\boldsymbol{x}(t), \boldsymbol{u}(t)) = x(t)\ln x(t) + x(t)u(t)$, and $\boldsymbol{x}_0 = 2$. The exact state and control variables are

$$x^*(t) = \exp\left(y^*(t)\right), \tag{6.1a}$$

$$u^*(t) = -\left(1 + \sqrt{2}\right) y^*(t), \tag{6.1b}$$

where

$$y^*(t) = (\ln 2)\exp\left(-\sqrt{2}t\right); \tag{6.1c}$$

see [44]. The exact cost function $J^* = (\ln 2)^2\left(\sqrt{2} + 1\right)/2 \approx 0.5799580911421756$ was rounded to 16 significant digits. Through the change of variables

$$z(t) = \ln x(t), \tag{6.2}$$

the IHOC described by Eqs (2.1)–(2.3) can be rewritten as an equivalent infinite-horizon linear-quadratic optimal control problem with $g(z(t), \boldsymbol{u}(t)) = \left(z^2(t) + u^2(t)\right)/2, f(z(t), \boldsymbol{u}(t)) = z(t) + u(t)$ and $z_0 = \ln 2$. We refer to the former and latter forms of the IHOC as Forms A and B, respectively. Form A of the example was previously solved in [44] by using LG- and LGR-PS methods and the three maps given by (3.2c) and (3.2d); the obtained NLPs were solved using the Sparse Nonlinear Optimizer (SNOPT) method [70, 71]. The problem was solved later in [46] by using the FRPM [46] and the mapping ζ_c:

$$\zeta_c(\tau) = \ln\left(\frac{4}{(1+\tau)^2}\right), \tag{6.3}$$

which maps the scaled left half-open interval $(-1, 1]$ to the descending time interval $(\infty, 0]$. The obtained NLP was also solved using SNOPT. Table 1 shows a comparison between the LGR- and LG-PS methods, the FRPM and the GGR-IPS12-EALMM. The LGR- and LG-PS methods and the GGR-IPS12-EALMM were performed using the same initial guesses $\tilde{x}(\tau) = 2$ and $\tilde{u}(\tau) = \tau\,\forall\tau \in [-1, 1]$. Notice how the GGR-IPS2-EALMM generally enjoy superior stability properties and achieve higher-order approximations in this example for $n = 5(5)30$ compared with the other approaches, except for the LG-PS method, where they both achieve the same order of accuracy at $n = 30$. It is interesting here to recognize how the GGR-IPS2-EALMM defeats the LG-PS method for $n = 5(5)25$, although the latter employs a Gauss quadrature that is more accurate than the GR quadrature used by the former. One may connect the success of the GGR-IPS2-EALMM here to many reasons, namely, (i) the clever change of variables given by Eq (6.2) that converts the NLP into a linear-quadratic optimal control problem which can be collocated more accurately; (ii) the integral form of the system dynamics allows for gaining more digits of accuracy via numerical quadratures, which are well known for their numerical stability; (iii) the application of the highly accurate built-in Algorithm B.2 to the current methods, as it applies the latest technology of SR interpolation; (iv) the parametric logarithmic map $T_{2,L}^{(\alpha)}$ that is favored over $T_{1,L}^{(\alpha)}$ for its slower growth and less sensitivity near $\tau = 1$; and (v) the map scaling parameter L, which permits faster convergence rates when "optimally" chosen. On the other hand, we observe that the errors of GGR-IPS12-EALMM generally decline gradually as the mesh grid size initially grow up to a certain limit, yet they bounce back beyond that limit as the mesh grid size continues to grow large, in agreement with the theoretical results of Section 5. It is interesting to see similar phenomena as in the control error profiles in [44] in the sense that (i) the control error plot of the LG-PS method does not appear as a (near) straight line in the shown log-scaled chart, but, rather, a convex-shaped curve, as it curves outward ([44, Figure 1(b)]), and (ii) the control error plot of the LGR-PS method suddenly increases at $n = 30$ much earlier before reaching the round-off plateau ([44, Figure 2(b)]). A closely related behavior to these observations can be found in [46], where we noticed that the control error history is also not linear in the shown log-scaled chart, but, rather, oscillate up and down at different levels again much earlier before reaching the round-off plateau ([46, Figure 5]). Another interesting remark lies in the smallest errors of the current methods; they were all recorded at/near $\alpha = 0.5$ with $\alpha \in \Upsilon_{0.5,0.6}^{G}$, while several optimal values of L were detected. Table 1 also shows the corresponding elapsed time (ET) to perform the GGR-IPS12-EALMM. The calculated execution times were measured multiple times and the shown data are the medians of the time measurements in seconds (s). Figures 7 and 8 show the plots of the exact state and control variables, in addition to their collocated solutions and absolute errors obtained by the GGR-IPS12 methods integrated with three NLP solvers using the same initial guesses set and several values of n, L and α. We can observe from the shown graphical data that the GGR-IPS2-EALMM generally achieves better accuracy and stability properties compared with the other methods.

Table 1. Uncertainty intervals of the smallest $MAE_{x,u}$ obtained by the LGR- and LG-PS methods in [44] and the FRPM [46] at the collocation points, and the corresponding smallest $MAE_{x,u}$ obtained by the GGR-IPS12-EALMM using the initial guesses $\tilde{x}(\tau) = 2$ and $\tilde{u}(\tau) = \tau\, \forall \tau \in [-1, 1]$. The ET values are also shown for the GGR-IPS12-EALMM. The errors and the ET values were rounded to five significant digits and three decimal digits, respectively.

Example 1					
	LGR-PS [44]	LG-PS [44]	FRPM [46]	GGR-IPS1-EALMM	GGR-IPS2-EALMM
		Form A		Form B	
n		$MAE_{x,u}$ uncertainty interval		$MAE_{x,u}/\alpha/L/ET$	$MAE_{x,u}/\alpha/L/ET$
5	(1e-03, 1e-02)	(1e-04, 1e-03)	–	5.3830e-03/0.6/2.25/0.379	4.2453e-05/0.5/3.5/0.237
10	(1e-06, 1e-05)	(1e-06, 1e-05)	(1e-05, 1e-04)	7.0615e-05/0.5/5.75/0.343	1.8735e-09/0.5/4.25/0.494
15	(1e-07, 1e-06)	(1e-07, 1e-06)	(1e-07, 1e-06)	6.0288e-07/0.5/9.25/0.451	1.9736e-09/0.5/5/0.529
20	(1e-08, 1e-07)	(1e-08, 1e-07)	(1e-08, 1e-07)	1.3181e-08/0.5/8.5/0.585	2.0583e-09/0.5/2.5/0.888
25	(1e-08, 1e-07)	(1e-08, 1e-07)	(1e-08, 1e-07)	2.4368e-08/0.5/2.25/1.160	1.6175e-09/0.5/3/1.066
30	(1e-08, 1e-07)	(1e-09, 1e-08)	(1e-08, 1e-07)	2.7958e-08/0.5/2.75/0.911	3.6927e-09/0.5/2/1.356

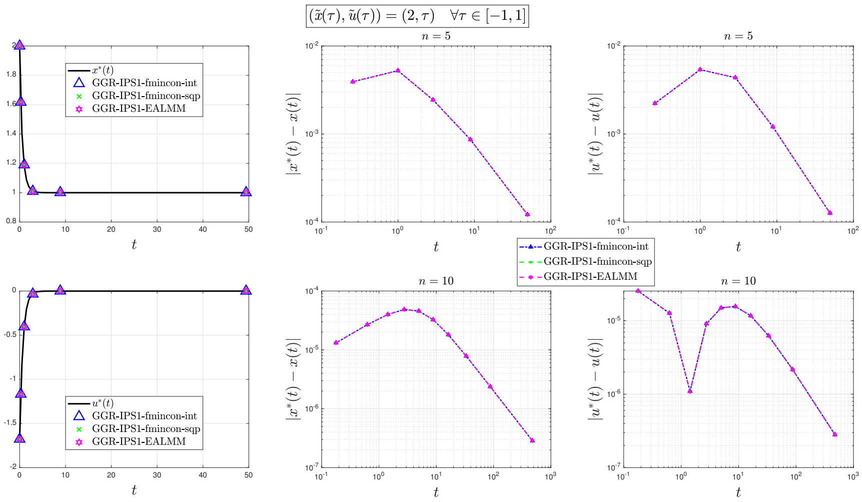

Figure 7. First column: the plots of the exact optimal state and control variables of Example 1 and the collocated solutions obtained by using the GGR-IPS1 method integrated with three distinct NLP solvers at the collocation points using $n = 5$ and the same initial guesses $\tilde{x}(\tau) = 2$ and $\tilde{u}(\tau) = \tau \, \forall \tau \in [-1, 1]$. The exact optimal state and control plots were generated using 101 linearly spaced nodes from 0 to 50. The middle and last columns show the corresponding plots of the absolute errors of the state and control variables in log-log scale using $n = 5, 10$ and the (L, α) ordered pairs as shown in Table 1.

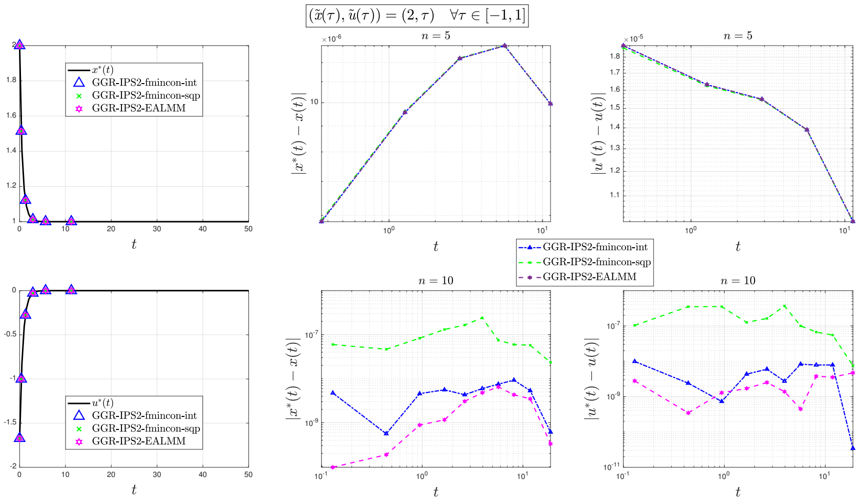

Figure 8. First column: the plots of the exact optimal state and control variables of Example 1 and the collocated solutions obtained by using the GGR-IPS2 method integrated with three distinct NLP solvers at the collocation points using $n = 5$ and the same initial guesses $\tilde{x}(\tau) = 2$ and $\tilde{u}(\tau) = \tau \, \forall \tau \in [-1, 1]$. The exact optimal state and control plots were generated using 101 linearly spaced nodes from 0 to 50. The middle and last columns show the corresponding plots of the absolute errors of the state and control variables in log-log scale using $n = 5, 10$ and the (L, α) ordered pairs as shown in Table 1.

Figures 9 and 10 show the $MAE_{x,u}$ of the GGR-IPS1-EALMM obtained by using $n = 10(10)50, \alpha = -0.4(0.1)2, L = 1(1)4$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}$. It is interesting to observe here by visual inspection that, when holding L fixed, the global minima of the error mesh surface plots occur near $\alpha = 0.5$ and the mesh surfaces rise up gradually as we move away, except when $\alpha \in \{-0.4, -0.3\}$, where sharp peaks may emerge suddenly for growing values of n. This suggests that Legendre polynomials seem to be an optimal choice among Gegenbauer basis polynomials when holding L fixed, while Gegenbauer polynomials associated with α values near -0.5 may cause numerical instability as n grows large. However, a different story emerges when the GGR-IPS2-EALMM is performed instead, as can be seen in Figures 11 and 12. Notice now that the errors appear to be monotonically decreasing for decreasing values of α when holding L fixed at 1 and 2, and that the error surface shoots up as n and α increases. Therefore, Gegenbauer polynomials with some negative α values seem optimal for relatively small values of L. Notice also that the sudden peaks observed before with the GGR-IPS1-EALMM in Figures 9 and 10 for $\alpha \in \{-0.4, -0.3\}$ and large n values disappear. A further array of error mesh surface plots of the GGR-IPS1-EALMM are shown in Figures 13 and 14 for $n = 10(10)50, \alpha = -0.2, 0, 0.5, 1, L = 0.5(0.5)6$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}$. While holding α fixed, there seems to be no general rule of thumb that can be drawn from the shown data. On the other hand, Figures 15 and 16 show the corresponding plots associated with the GGR-IPS2-EALMM, where the errors are very similar and can be clearly seen to surge as $L \to 0$, especially when $\alpha = 1$, but remain relatively small for $L = 2(0.5)6$.

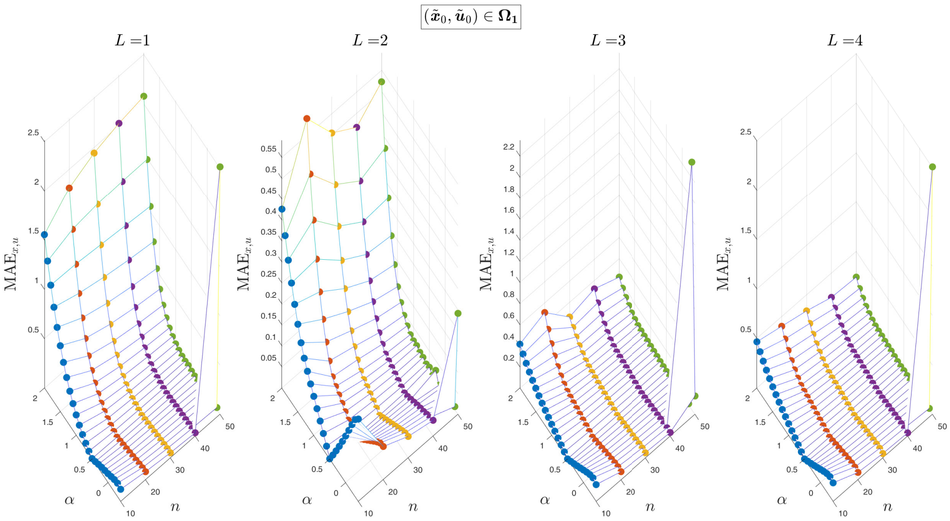

Figure 9. $MAE_{x,u}$ of the GGR-IPS1-EALMM at 101 linearly spaced nodes between 0 and 10, as obtained by using $n = 10(10)50, \alpha = -0.4(0.1)2, L = 1(1)4$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$.

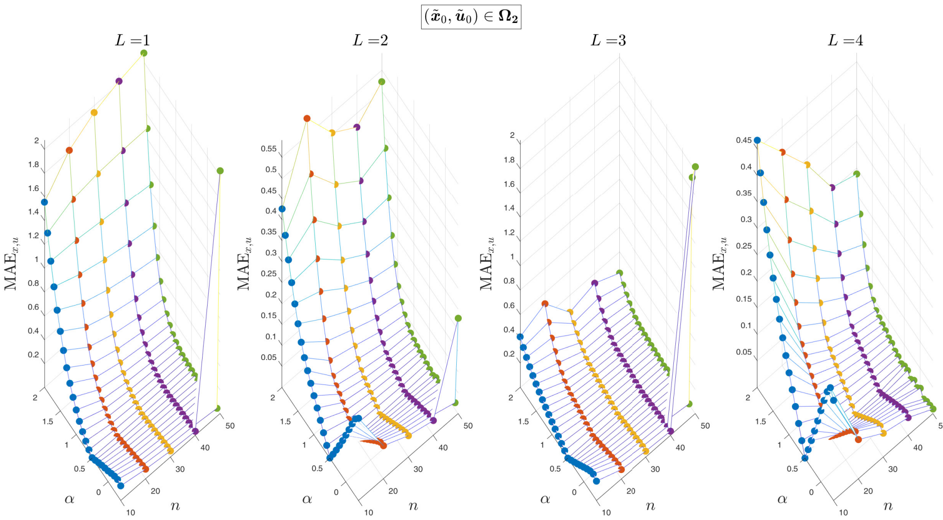

Figure 10. $MAE_{x,u}$ of the GGR-IPS1-EALMM at 101 linearly spaced nodes between 0 and 10, as obtained by using $n = 10(10)50, \alpha = -0.4(0.1)2, L = 1(1)4$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_2$.

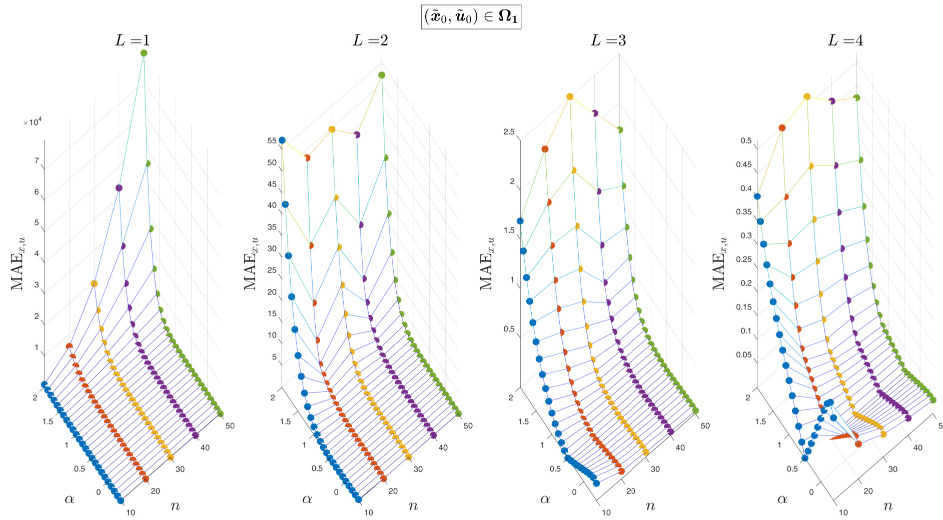

Figure 11. $MAE_{x,u}$ of the GGR-IPS2-EALMM at 101 linearly spaced nodes between 0 and 10, as obtained by using $n = 10(10)50, \alpha = -0.4(0.1)2, L = 1(1)4$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$.

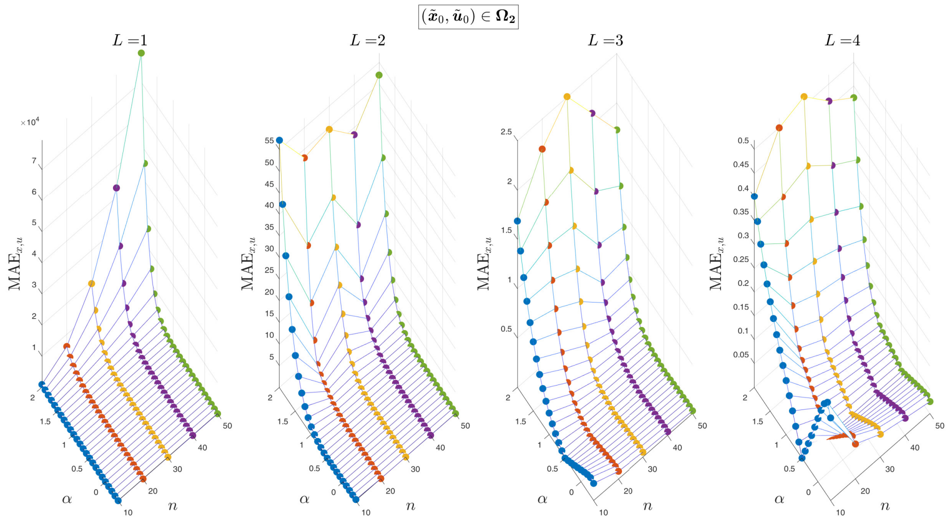

Figure 12. $MAE_{x,u}$ of the GGR-IPS2-EALMM at 101 linearly spaced nodes between 0 and 10, as obtained by using $n = 10(10)50, \alpha = -0.4(0.1)2, L = 1(1)4$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_2$.

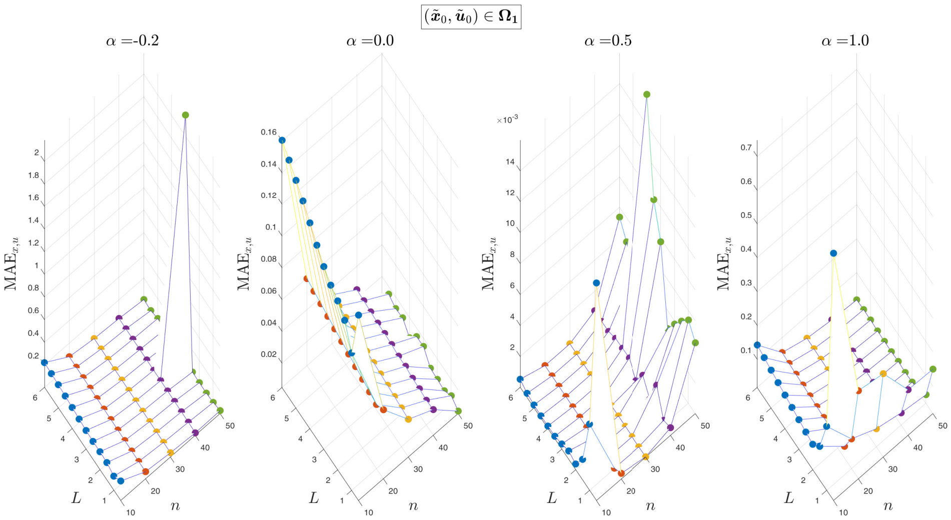

Figure 13. $MAE_{x,u}$ of the GGR-IPS1-EALMM at 101 linearly spaced nodes between 0 and 10, as obtained by using $n = 10(10)50, \alpha = -0.2, 0, 0.5, 1, L = 0.5(0.5)6$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$.

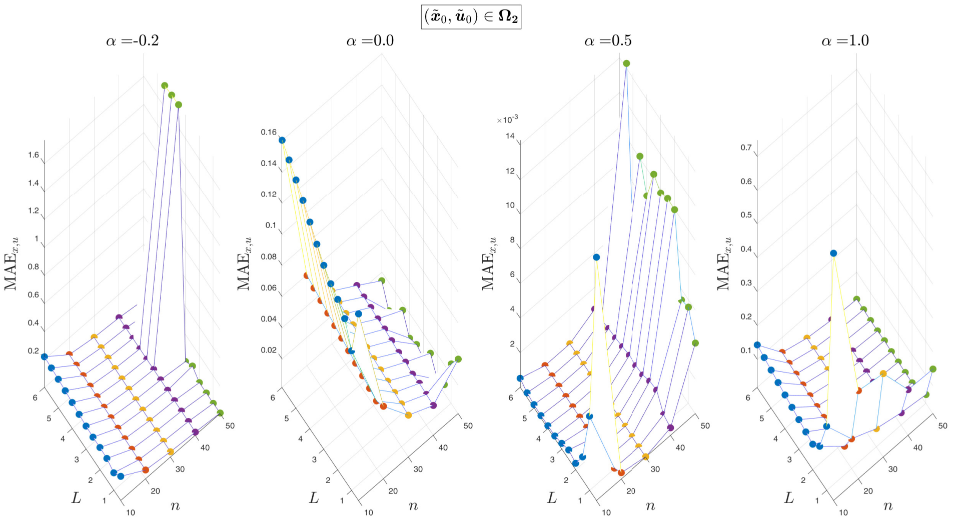

Figure 14. $MAE_{x,u}$ of the GGR-IPS1-EALMM at 101 linearly spaced nodes between 0 and 10, as obtained by using $n = 10(10)50, \alpha = -0.2, 0, 0.5, 1, L = 0.5(0.5)6$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_2$.

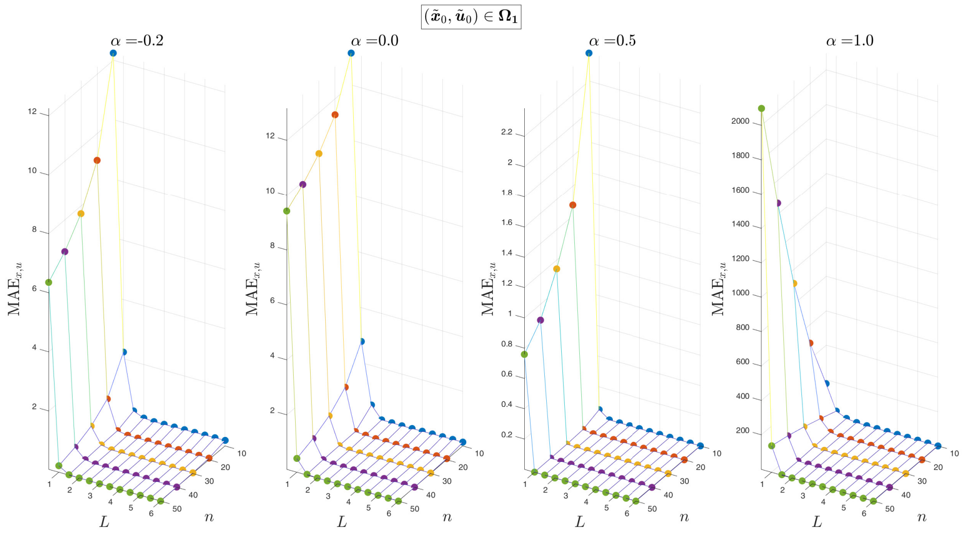

Figure 15. $MAE_{x,u}$ of the GGR-IPS2-EALMM at 101 linearly spaced nodes between 0 and 10, as obtained by using $n = 10(10)50, \alpha = -0.2, 0, 0.5, 1, L = 0.5(0.5)6.5$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$.

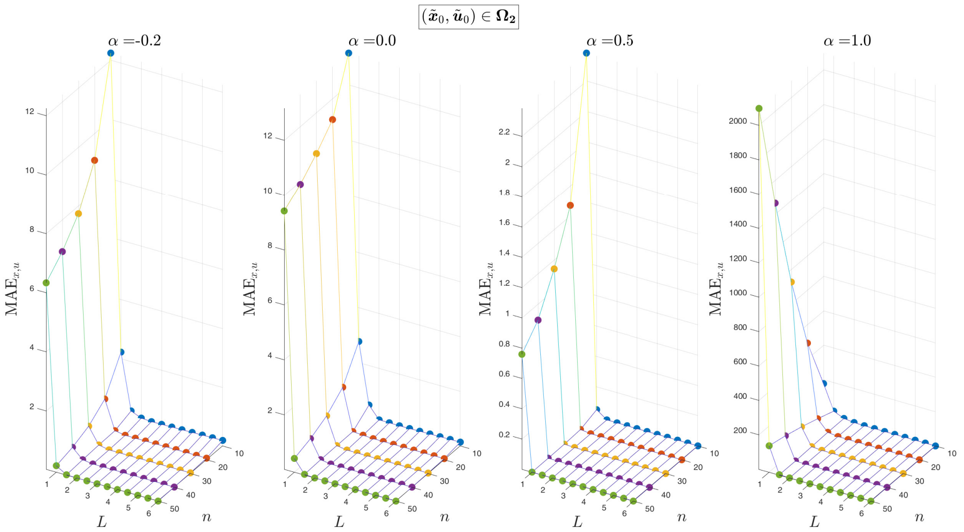

Figure 16. $MAE_{x,u}$ of the GGR-IPS2-EALMM at 101 linearly spaced nodes between 0 and 10, as obtained by using $n = 10(10)50, \alpha = -0.2, 0, 0.5, 1, L = 0.5(0.5)6.5$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_2$.

Figures 17 and 18 show comparisons of the number of iterations required by the GGR-IPS12 methods when combined with three distinct NLP solvers using $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}$ and several L and α values. Clearly, the integration of the GGR-IPS12 methods with the EALMM leads to a drastic reduction in the number of iterations in all cases. In fact, while the GGR-IPS12-EALMM often converged in only four/five iterations, other methods usually require many more iterations to converge; for example, the GGR-IPS12 methods combined with fmincon-int and fmincon-sqp took more than 200 iterations to converge to the solutions of the problem when $(n, L, \alpha) = (48, 1, -0.2)$ and starting with any initial guess $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}$.

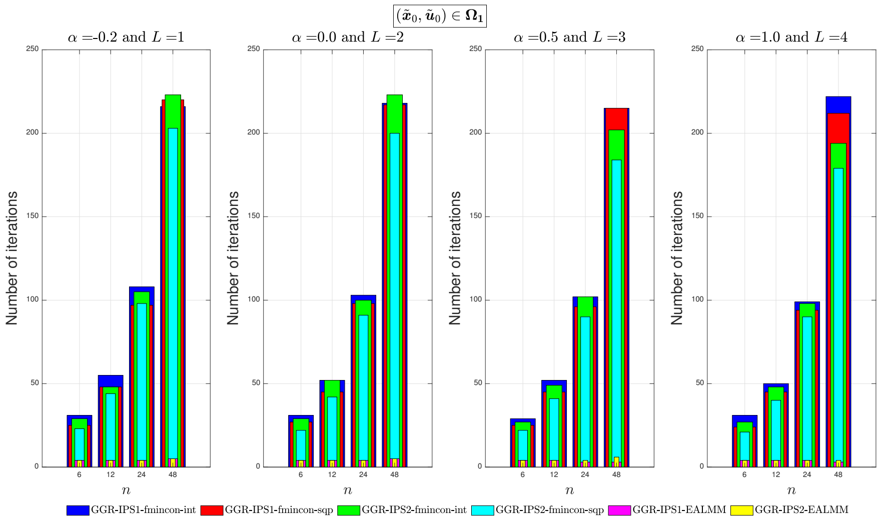

Figure 17. Number of iterations required by the GGR-IPS12 methods performed with three distinct NLP solvers versus n for $(\alpha, L) = (-0.2, 1), (0, 2), (0.5, 3), (1, 4)$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$.

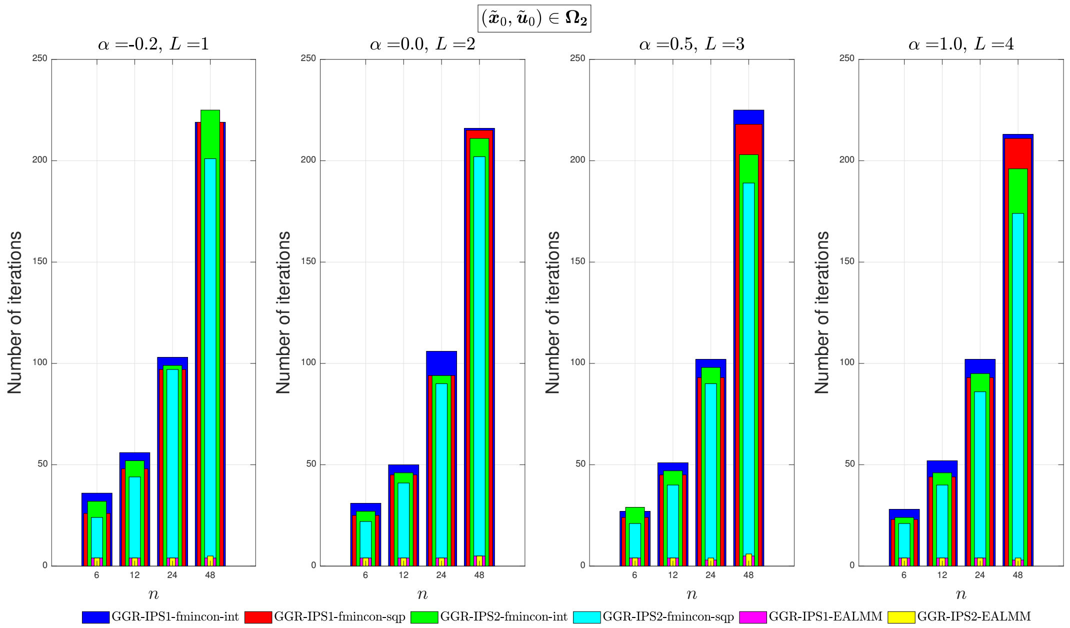

Figure 18. Number of iterations required by the GGR-IPS12 methods performed with three distinct NLP solvers versus n for $(\alpha, L) = (-0.2, 1), (0, 2), (0.5, 3), (1, 4)$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \mathbf{\Omega}_2$.

Table 2 shows the approximate cost function values obtained by the GGR-IPS2-EALMM for several parameter values. The fastest convergence was recorded at $\alpha = 0.5$ in all cases with $J \approx J_{16} = 0.579958091127$, which is in agreement with J to 10 significant digits. It is interesting to see through the tabulated data how the Gegenbauer polynomials with $\alpha \in \{-0.4, 0.25\}$ exhibit faster convergence rates than the Chebyshev polynomials (when $\alpha = 0$), capturing four correct significant digits as early as $n = 12$, whereas the Chebyshev polynomials are still lagging behind by one digit even when n increases by two units. Gegenbauer polynomials with $\alpha = -0.2$ also performed better than the Chebyshev polynomials for $n \in \{14, 16\}$, while the poorest stability was that of Gegenbauer polynomials with $\alpha = 2$, scoring only one correct significant digit in all cases. Table 3 shows the smallest $MAE_{x,u}$ obtained by the GGR-IPS12-EALMM among the recorded errors for the parameter values $n = 10(10)80, \alpha = -0.4(0.1)2, L = 0.25(0.25)10$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \mathbf{\Omega}$. The table shows the capacity of the GGR-IPS12-EALMM to achieve improved near-optimal solutions for increasing values of n within a small/medium range of mesh grid size; however, the accuracy deteriorates beyond a certain limit as the mesh grid size grows larger, in agreement with the theoretical results proven in Section 5. The GGR-IPS2-EALMM is clearly superior to the GGR-IPS1-EALMM in terms of accuracy in all cases, and it is interesting here to see how the GGR-IPS1-EALMM diverges faster than the GGR-IPS2-EALMM for growing mesh sizes, as indicated earlier by the divergence analysis presented in Section 5. The best approximations obtained experimentally by the GGR-IPS2-EALMM were recorded at/near $\alpha = 0.5$ with $\alpha \in \Upsilon^G_{0.4,0.6}$. The smallest errors of the GGR-IPS1-EALMM were also recorded at/near $\alpha = 0.5$ for $n = 5(5)55$ with $\alpha \in \Upsilon^G_{0.4,0.6}$; however, the algorithm tends to favor larger positive α values beyond $n = 55$, where we noticed the travel of the right boundary, α_c^+, of $\Upsilon^G_{0.4,\alpha_c^+}$ rightward from $\alpha_c^+ = 0.6$ into $\alpha_c^+ = 1.8$ as n reaches 80. This is no surprise. In fact, recall that $T^{(\alpha)}_{1,L}$ increases monotonically for decreasing values of α while holding n and L fixed, and we can observe from Figure 1 that the mapping escalates wildly as we continue decreasing the α values. The byproduct of this behavior is that increasing the α values moves the collocation points associated with large values of t leftward, relocating them closer to regions where the solution changes rapidly. This graphical interpretation is consistent with the fact that the interior GGR points move monotonically toward the center of the interval $(-1, 1)$ as the parameter α increases [51, 56]. From another perspective, the leftward movement of the collocation points near the right boundary $\tau = 1$ mitigates the effect of the ill-conditioning of $T'_{1,L}$ for arguments near 1, as $T'_{1,L}$ is evaluated at mesh points that are gradually departing the vicinity of $\tau = 1$. This argument adds more tenability to the cause of using Gegenbauer polynomials as a basis polynomials for numerical collocations of FHOCIs obtained from IHOCs via $T_{1,L}^{(\alpha)}$ or $T_{2,L}^{(\alpha)}$ in the sense that, while Chebyshev and Legendre polynomials cease to downgrade the errors as the mesh grid size grows large, Gegenbauer polynomials have the additional advantage of being able to alleviate the growth rates of both $T_{1,L}^{(\alpha)}$ and $T_{2,L}^{(\alpha)}$ by increasing the α value whenever we wish, while sustaining the luxury of being able to apply either Chebyshev or Legendre polynomials. If we now turn our attention to the recorded L values in the table, we can quickly spot that the smallest computed errors initially span a wide range of numerically optimal L values; however, the solvers ultimately have a bias toward smaller values of L in an attempt to damp the error in agreement with the forward error analysis presented in Section 5. Notice that the numerically optimal L value for the GGR-IPS2-EALMM stays at 0.75 for $n = 55(5)80$, while the corresponding values for the GGR-IPS1-EALMM occur at the smallest feasible L value among the input range of experimental data. One may attribute this peculiar behavior of the solvers to the fact that the logarithmic map $T_{2,L}^{(\alpha)}$ increases at a much slower rate than that of the algebraic map $T_{1,L}^{(\alpha)}$; see Figures 1 and 2.

Table 2. Approximate cost function values obtained by the GGR-IPS2-EALMM for $(n, L) = (6, 1), (8, 2), (10, 3), (12, 4), (14, 5), (16, 6)$ and $\alpha = -0.4, -0.2, 0, 0.25, 0.5, 1, 2$. All approximations were rounded to 12 significant digits.

Example 1								
n	L	$\alpha = -0.4$	$\alpha = -0.2$	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 2$
6	1	0.579809073360	0.579627701619	0.579622685738	0.579789248084	0.579949642114	0.577727846201	0.503481602677
8	2	0.579848669619	0.579713782304	0.579730070685	0.579859689930	0.579958090977	0.578484510558	0.522640769897
10	3	0.579893894832	0.579797498143	0.579802788432	0.579889201985	0.579958091142	0.578845602933	0.530047022566
12	4	0.579918900010	0.579850348844	0.579850444796	0.579908959905	0.579958091143	0.579089227180	0.534636030121
14	5	0.579933051361	0.579883424648	0.579881314693	0.579922126600	0.579958091151	0.579263254798	0.538003914873
16	6	0.579941397724	0.579904638260	0.579901719586	0.579931066922	0.579958091127	0.579391311257	0.540689754740

Table 3. $MAE_{x,u}$ of the GGR-IPS12-EALMM obtained by using $n = 5(5)80$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}$. All approximations were rounded to five significant digits.

	Example 1											
	GGR-IPS1-EALMM						GGR-IPS2-EALMM					
	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$			$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_2$			$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$			$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_2$		
n	α	L	$MAE_{x,u}$	α	L	$MAE_{x,u}$	α	L	$MAE_{x,u}$	α	L	$MAE_{x,u}$
5	0.6	2.25	5.3830e-03	0.6	2.25	5.3830e-03	0.5	3.5	4.2456e-05	0.5	3.5	4.2458e-05
10	0.5	5.75	6.9439e-05	0.5	5.75	7.0620e-05	0.5	5.25	8.6420e-09	0.5	4.75	9.8263e-09
15	0.5	7	6.3879e-07	0.5	7	6.7410e-07	0.5	4	6.1199e-09	0.5	3.25	5.7262e-09
20	0.5	10	2.3194e-07	0.5	9	1.4376e-07	0.5	2.25	6.7398e-09	0.5	2.5	8.6293e-09
25	0.5	9	1.7060e-07	0.5	5.75	1.5636e-07	0.5	2.25	4.4629e-08	0.5	5.75	3.7128e-08
30	0.5	2	4.9161e-07	0.5	3.25	2.9685e-07	0.5	1.75	6.4234e-08	0.5	1.75	9.7104e-08
35	0.5	0.75	2.6309e-06	0.5	0.5	5.3234e-06	0.5	1.75	1.0386e-07	0.5	1.75	9.4232e-08
40	0.5	0.25	6.7082e-05	0.5	0.25	1.8574e-05	0.5	1.5	4.7475e-07	0.5	2	2.6221e-07
45	0.5	1	7.2641e-05	0.5	4.25	1.3092e-04	0.5	1.25	5.1101e-07	0.5	1.75	2.4520e-07
50	0.4	1.5	9.7024e-04	0.4	3.25	1.0915e-03	0.5	1	8.6070e-06	0.5	1.25	6.6680e-06
55	0.6	4.75	1.1300e-03	0.6	7.25	1.1785e-03	0.5	0.75	1.8330e-05	0.5	0.75	1.8329e-05
60	1	0.5	1.9681e-02	0.7	0.25	1.6967e-02	0.5	0.75	7.0032e-05	0.5	0.75	1.4334e-04
65	1.2	0.25	2.2391e-02	1.2	0.25	2.2419e-02	0.5	0.75	3.1392e-04	0.5	0.75	2.1468e-04
70	1.4	0.25	5.8462e-02	1.3	0.25	7.7659e-02	0.5	0.75	1.0067e-03	0.4	0.75	8.0880e-04
75	1.4	0.25	1.5099e-01	1.4	0.25	1.5681e-01	0.6	0.75	8.6516e-04	0.5	0.75	8.9362e-04
80	1.8	0.25	3.2058e-01	1.8	0.25	2.7602e-01	0.4	0.5	8.3497e-04	0.4	0.75	6.7363e-04

Example 2. Consider the IHOC given by Eqs (2.1)–(2.3) with $g\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right) = x_1^2(t) + x_2^2(t)/2 + u^2(t)/4, \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right) = [x_2(t), 2x_1(t) - x_2(t) + u(t)]^\top$ and $\boldsymbol{x}_0 = [-4, 4]^\top$. The exact state and control variables are

$$\boldsymbol{x}^*(t) = \exp\left(\mathcal{M}t\right)\boldsymbol{x}(0), \tag{6.4a}$$

$$u^*(t) = -\boldsymbol{K}\boldsymbol{x}^*(t), \tag{6.4b}$$

where

$$\mathcal{M} = \begin{bmatrix} 0 & 1 \\ -2.82842712474619 & -3.557647291327851 \end{bmatrix}, \tag{6.4c}$$

$$\boldsymbol{K} = [4.828427124746193; 2.557647291327851]\,; \tag{6.4d}$$

see [45, 55, 72]. This example is a linear-quadratic regulator problem with an optimal cost functional value $J^* = 19.85335656362790$, rounded to 16 significant digits, as obtained in MATLAB by using the Symbolic Math Toolbox. Figure 19 shows the plots of the exact optimal state and control variables and their approximations obtained through GGR-IPS2-EALMM using some parameter values. Table 4 shows the $MAE_{x,u}$ of the LGR-PS method in [45] and the smallest corresponding $MAE_{x,u}$ and AE_J pairs of the GGR-IPS2-EALMM at the collocation points, as obtained by using $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}$ and several values of n. The table also shows the corresponding ET taken to perform the GGR-IPS2-EALMM. The GGR-IPS2-EALMM proves again to be superior in terms of accuracy for a small range of mesh sizes, as it converges rapidly to near-optimal solutions at a much higher rate than that in [45]. However, the superb accuracy of the method starts to decline when n grows larger, as anticipated earlier. Notice again here that the best accuracy in all cases was recorded at $\alpha = 0.5$ with SRCIC $\Upsilon_{0.5,0.5}^G = \{0.5\}$.

Table 4. Uncertainty intervals of the smallest $MAE_{x,u}$ obtained by employing the method in [45] at the collocation points and the corresponding smallest $MAE_{x,u}$ and AE_J pairs obtained by using the GGR-IPS2-EALMM with $n = 4(5)34$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}$. The ET values are also shown for the GGR-IPS2-EALMM. The errors and the ET values were rounded to five significant digits and three decimal digits, respectively.

Example 2			
	Method of Garg et al. [45]	GGR-IPS2-EALMM	
		$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_2$
$n+1$	$MAE_{x,u}$ uncertainty interval	$MAE_{x,u}/AE_J/\alpha/L/ET$	$MAE_{x,u}/AE_J/\alpha/L/ET$
5	(1e-01,1)	1.6959e-03/1.7032e-05/0.5/1.75/0.422	1.6959e-03/1.7032e-05/0.5/1.75/1.988
10	(1e-02,1e-01)	9.4155e-08/1.7870e-12/0.5/2.5/3.532	8.5595e-07/9.6705e-12/0.5/3.25/3.452
15	(1e-04, 1e-02)	2.0165e-08/4.9489e-12/0.5/5.5/1.447	1.2501e-08/2.1316e-13/0.5/2.5/1.631
20	(1e-04, 1e-03)	8.2183e-09/1.6485e-12/0.5/3.5/1.866	6.2243e-09/1.0040e-11/0.5/2.5/3.695
25	(1e-05, 1e-04)	1.7564e-07/3.6451e-12/0.5/3.5/1.103	8.4676e-08/1.7483e-11/0.5/3/1.231
30	(1e-06, 1e-05)	1.1714e-06/3.1175e-11/0.5/1.75/6.212	2.4535e-06/4.9347e-12/0.5/1.75/4.942
35	(1e-06, 1e-05)	6.9522e-06/9.9437e-11/0.5/1.5/7.971	4.5463e-06/1.0522e-10/0.5/1.5/1.592

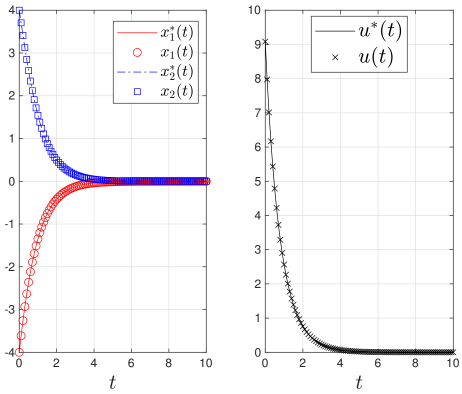

Figure 19. Exact optimal states and control of Example 2, as well as their collocated approximations obtained by applying GGR-IPS2-EALMM on the interval $[0, 10]$ using $n = 9, \alpha = 0.5, L = 2.5$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$. All figures were generated using 101 linearly spaced nodes from 0 to 10.

Another comparison between GGR-IPS2-fmincon-int, GGR-IPS2-fmincon-sqp and the transformed LGR method in [47] is shown in Table 5. We can clearly see that the former two methods generally yield smaller AE_J values. The rise and fall of accuracy as the mesh size grows is again peculiar in the observed approximations, which is in agreement with the presented divergence analysis in Section 5. Remarkably, a match with the exact J^* to full machine precision was recorded as early as $n = 20$, indicating an exceedingly accurate numerical scheme with exponential convergence for coarse meshes. All of the smallest errors reported by the current methods occurred at $\alpha = 0.5$, except for $n \in \{90, 100\}$, where collocations at $\alpha = 0$ and 0.8 furnished higher accuracy. On the other hand, the rounded errors in [47] decay to 1.35×10^{-09} as soon as $n = 30$ when using the algebraic map, but they cease to vary any further for $n = 40(10)100$.

Table 6 shows a third comparison between GGR-IPS2-fmincon-int, GGR-IPS2-fmincon-sqp and the recent method of Mamehrashi and Nemati [73], which uses Laguerre functions and the Ritz spectral method. All errors produced by the current methods were smaller than those obtained in [73] under the condition of using the same number of approximation terms; notably, a large difference of nine and ten orders of magnitude was observed in favor of the current methods using as small as $n = 10$.

Table 5. AE_J of the method in [47] and the smallest AE_J obtained by GGR-IPS2-fmincon-int and GGR-IPS2-fmincon-sqp using $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}$. The ET values are also shown for the latter two methods. The errors and the ET values were rounded to five significant digits and three decimal digits, respectively.

Example 2						
	Method of Shahini and Mehrpouya [47]		GGR-IPS2			
	Algebraic map $T_{1,L}^{(\alpha)}$	Logarithmic map $T_{2,L}^{(\alpha)}$	fmincon-int		fmincon-sqp	
			$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_2$	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_2$
n	AE_J		$AE_J/\alpha/L/ET$	$AE_J/\alpha/L/ET$	$AE_J/\alpha/L/ET$	$AE_J/\alpha/L/ET$
10	4.82e-05	3.92e-05	5.3291e-14/0.5/2.5/0.169	5.3291e-14/0.5/2.5/0.137	1.0658e-14/0.5/4.25/0.075	7.1054e-15/0.5/2.5/0.076
20	4.88e-09	1.76e-06	0/0.5/6/0.513	0/0.5/5.25/0.488	0/0.5/5.75/0.340	0/0.5/5.75/0.364
30	1.35e-09	2.73e-07	7.1054e-15/0.5/3/1.109	1.4211e-14/0.5/5.25/1.055	1.0658e-14/0.5/2.5/0.971	1.7764e-14/0.5/5.5/0.895
40	1.35e-09	7.24e-08	1.0658e-14/0.5/5.75/2.201	3.1974e-14/0.5/5.25/2.136	1.0303e-13/0.5/6/1.8	1.2079e-13/0.5/5/1.752
50	1.35e-09	2.63e-08	8.8818e-14/0.5/10/3.748	1.1013e-13/0.5/2.5/3.852	5.1514e-13/0.5/5/2.737	5.9686e-13/0.5/5/2.645
60	1.35e-09	1.19e-08	1.7053e-13/0.5/2.5/5.711	4.0146e-13/0.5/10/5.485	8.3844e-13/0.5/5.5/3.758	1.0409e-12/0.5/5.75/3.702
70	1.35e-09	6.45e-09	2.7001e-13/0.5/5/8.123	4.0501e-13/0.5/10/7.857	1.2967e-12/0.5/10/5.457	1.7337e-12/0.5/5/5.229
80	1.35e-09	4.06e-09	8.2423e-13/0.5/5.5/10.965	8.3134e-13/0.5/5.5/10.923	1.7977e-12/0.5/10/6.952	2.7676e-12/0.5/2.5/6.863
90	1.35e-09	2.90e-09	1.1072e-10/0/5.25/11.269	2.1283e-09/0/2.5/11.371	1.4021e-09/0/3.5/7.975	1.4747e-10/0/2.5/7.933
100	1.35e-09	2.29e-09	9.5391e-08/0.8/4/12.601	1.9779e-07/0.8/3.5/12.499	5.3783e-08/0.8/4/7.671	1.6219e-07/0.8/3.75/7.887

Table 6. AE_J of the method in [73] and the smallest AE_J obtained by GGR-IPS2-fmincon-int and GGR-IPS2-fmincon-sqp using $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}$. The ET values are also shown for the latter two methods. The errors and the ET values were rounded to five significant digits and three decimal digits, respectively.

Example 2					
	Method of Mamehrashi and Nemati [73]	GGR-IPS2			
		fmincon-int		fmincon-sqp	
		$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_2$	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_2$
n	AE_J	$AE_J/\alpha/L/ET$	$AE_J/\alpha/L/ET$	$AE_J/\alpha/L/ET$	$AE_J/\alpha/L/ET$
1	8.0868e-02	3.5950e-03/1.6/3.75/0.032	3.5950e-03/1.6/3.75/0.031	3.5950e-03/1.6/3.75/0.009	3.5950e-03/1.6/3.75/0.008
3	2.5290e-02	1.3845e-04/0.5/3/0.035	1.3845e-04/0.5/3/0.035	1.3845e-04/0.5/3/0.015	1.3845e-04/0.5/3/0.014
10	9.5436e-05	5.3291e-14/0.5/2.5/0.169	5.3291e-14/0.5/2.5/0.137	1.0658e-14/0.5/4.25/0.075	7.1054e-15/0.5/2.5/0.076

Example 3. Consider the IHOC given by (2.1)–(2.3) with

$$g\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right) = \frac{1}{2}\left[\boldsymbol{x}^\top(t)\mathbf{Q}\boldsymbol{x}(t) + \boldsymbol{u}^\top(t)\mathbf{R}\boldsymbol{u}(t)\right], \tag{6.5}$$

$$\boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\boldsymbol{x}(t) + \begin{bmatrix} -x_1^3(t) + x_2^2(t) \\ x_1(t)x_2(t) + x_2^3(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\boldsymbol{u}(t) \tag{6.6}$$

and $\boldsymbol{x}_0 = [0, 0.8]^\top$, where $\mathbf{Q} = \mathbf{R} = \mathbf{I}_2$. This problem was solved earlier in [74] by using a piecewise Adomian decomposition method, and again recently in [73] by using Laguerre functions and the Ritz spectral method. Table 7 shows comparisons between these two methods and the GGR-IPS2-EALMM with $\alpha = 0.5, L = 1$ and $n = 1(1)10$. Because of the agreement of the approximations obtained by the GGR-IPS2-EALMM for $n = 9$ and 10, we could reasonably expect the value of J_{10} to be accurate to the places listed. It is remarkable here to notice that J_1 agrees with J_{10} to three significant digits under the condition of using as little as two collocation nodes, which reflects the swift convergence rate achieved by the current method for coarse meshes. Notice also that the obtained J_5 value is smaller than the corresponding approximate optimal cost function values obtained using both rival methods. The state and control numerical solutions obtained by the GGR-IPS2-EALMM are shown in Figure 20.

Table 7. Comparisons of the approximate optimal cost function values. The approximations obtained using the GGR-IPS2-EALMM were rounded to seven decimal digits. The rounded ET values to three decimal digits are also shown for the GGR-IPS2-EALMM.

Example 3

	Method of Nik et al. [74]	Method of Mamehrashi and Nemati [73]			
		$n = 1$	$n = 2$	$n = 4$	$n = 5$
Approximate optimal J	0.5350	0.31129	0.21346	0.2109608	0.2109407

	GGR-IPS2-EALMM using $\alpha = 0.5, L = 1, (\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$									
	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
Approximate optimal J	0.2142078	0.2103731	0.2109176	0.2109341	0.2109368	0.2109375	0.2109379	0.2109380	0.2109381	0.2109381
ET	0.346	0.778	1.262	1.108	1.415	1.085	1.510	2.053	2.371	1.848

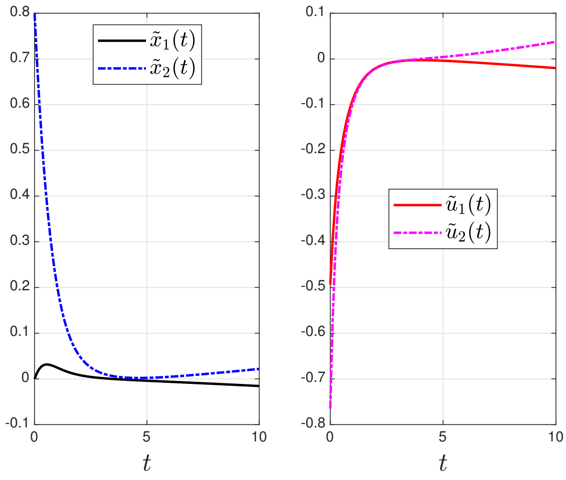

Figure 20. Collocated state and control solutions of Example 3 on the interval $[0, 10]$; the plots were obtained by employing the GGR-IPS2-EALMM with $n = 5, \alpha = 0.5, L = 1$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \boldsymbol{\Omega}_1$.

7. Conclusions and future work

Direct IPS methods for solving IHOCS using the logarithmic mapping $T_{2,L}^{(\alpha)}$ and the developed SR interpolation and quadrature formulas can produce excellent approximations to the optimal state and control variables for relatively small/medium mesh grids. However, this class of methods often suffer from numerical instability for fine meshes when endowed with any of the parametric maps $T_{i,L}^{(\alpha)}, i = 1, 2$; therefore, as the mesh size grows, they are not as useful as one might hope for computing the optimal state and control trajectories to within high precision. In fact, it has been shown in the current paper that two sources of difficulty arise in handling the horizon in IHOCs using a domain transformation that maps the infinite horizon to the finite horizon $[-1, 1)$ through the algebraic and logarithmic maps

$T_{i,L}^{(\alpha)}, i = 1,2$: (i) the exponential growth of the mappings surface slopes near the right boundary $\tau = 1$, which increases the truncation errors produced in the FHOCI discretization without bounds as $\tau \to 1$; and (ii) despite the fact that both mappings $T_{i,L}^{(\alpha)}, i = 1,2$ have a singularity at $\tau = 1$ and we actually never evaluate them at the singularity, since the GGR collocation points are strictly less than 1, their derivatives are sensitive to input data errors for arguments near $\tau = 1$; thus, both NLP1 and NLP2 are ill-conditioned for $\tau \approx 1$. These theoretical facts, as well as the observed empirical data, are considerable reasons to say that typical direct spectral/PS and IPS methods based on classical Jacobi polynomials and the parametric maps $T_{i,L}^{(\alpha)}, i = 1,2$ are foreseen to diverge as the mesh size grows large if the computations are carried out using floating-point arithmetic and the discretizations use a single mesh grid, regardless of whether they are of Gauss type or equally spaced, which is a result that contradicts the convergence claim of such methods using Legendre polynomials made in [44].

While Gegenbauer polynomials associated with certain nonpositive α values are well suited for FGGR-based polynomial interpolations in Lagrange basis form over fine meshes, as shown in [24]; this paper asserts that Gegenbauer polynomials associated with certain nonnegative α values are more apt for GGR-based SR interpolations over fine meshes. Moreover, for coarse mesh grids, Legendre polynomials are particularly (near) optimal basis polynomials for GGR-based SR collocations of FHOCIs, as argued in Section 4.1 and sustained through numerical simulations. On the other hand, Gegenbauer polynomials associated with certain positive values of $\alpha \in (1/2, 2]$ are optimal for IHOCI collocations over fine mesh grids, as they can largely slow down the exponential growth of both parametric maps $T_{i,L}^{(\alpha)}, i = 1,2$, and their associated GGR collocation points are less dense near $\tau = 1$; thus, the sensitivity of computing $T'_{i,L}, i = 1,2$ at arguments near $\tau = 1$ is significantly attenuated. The paper also shows that the parametric map $T_{1,L}^{(\alpha)}$ is more severely sensitive for $\tau \approx 1$ than $T_{2,L}^{(\alpha)}$, and that the family $\left\{\left(T_{1,L}^{(\alpha)}\right)^m\right\}_{m=0}^{\infty}$ grows faster than $\left\{\left(T_{2,L}^{(\alpha)}\right)^m\right\}_{m=0}^{\infty}$ as $\tau \to 1$; therefore, $T_{2,L}^{(\alpha)}$ is more apt for the domain transformation of IHOCs than $T_{1,L}^{(\alpha)}$ for collocation points of Gauss/GR type.

It is worthy to mention that direct IPS methods based on the proposed Gegenbauer SR collocation can exhibit faster convergence rates for coarse meshes by regulating the map scaling parameter L and the Gegenbauer parameter α. In light of the stability analysis conducted in Section 4.1, GGR-based SR collocations of well-conditioned problems are generally endorsed for α values within/near the SRCIC $\Upsilon_{1/2,1}^{G}$; the current study also supports this rule of thumb for IHOCs that are converted into FHOCIs through the use of parametric maps $T_{i,L}^{(\alpha)}, i = 1,2$ and then collocated at relatively coarse mesh grids. On the other hand, a rule of thumb for choosing the optimal map scaling parameter value, L^*, based on the derived error bounds in Section 5 and extensive numerical simulations performed in Section 6 suggests to choose L within the range $(0,1)$ for large mesh grids. However, the question of how can we find L^* remains open for coarse meshes. An interesting direction for future work may involve a study of new mappings with slower growth rates and derivatives with less sensitivity to input data errors. Another interesting direction is to extend the current work to handle IHOCs subject to boundary value problems.

Conflict of interest

The authors declare no conflicts of interest regarding this paper.

References

1. S. A. Orszag, Accurate solution of the Orr–Sommerfeld stability equation, *J. Fluid Mech.*, **50** (1971), 689–703. https://doi.org/10.1017/S0022112071002842
2. G. S. Patterson, S. A. Orszag, Spectral calculations of isotropic turbulence: efficient removal of aliasing interactions, *Phys. Fluids*, **14** (1971), 2538–2541. https://doi.org/10.1063/1.1693365
3. W. Kang, N. Bedrossian, Pseudospectral optimal control theory makes debut flight, saves NASA $ 1M in under three hours, *SIAM News*, **40** (2007), 1–3.
4. Z. Liu, S. Li, K. Zhao, Extended multi-interval Legendre-Gauss-Radau pseudospectral method for mixed-integer optimal control problem in engineering, *Int. J. Syst. Sci.*, **52** (2021), 928–951. https://doi.org/10.1080/00207721.2020.1849862
5. K. T. Elgindy, A high-order embedded domain method combining a Predictor-Corrector-Fourier-Continuation-Gram method with an integral Fourier pseudospectral collocation method for solving linear partial differential equations in complex domains, *J. Comput. Appl. Math.*, **361** (2019), 372–395. https://doi.org/10.1016/j.cam.2019.03.032
6. M. Nazari, M. Nazari, M. H. N. Skandari, Pseudo-spectral method for controlling the drug dosage in cancer, *IET Syst. Biol.*, **14** (2020), 261–270. https://doi.org/10.1049/iet-syb.2020.0054
7. K. T. Elgindy, B. Karasözen, High-order integral nodal discontinuous Gegenbauer-Galerkin method for solving viscous Burgers' equation, *Int. J. Comput. Math.*, **96** (2019), 2039–2078. https://doi.org/10.1080/00207160.2018.1554860
8. B. Fornberg, D. M. Sloan, A review of pseudospectral methods for solving partial differential equations, *Acta Numer.*, **3** (1994), 203–267. https://doi.org/10.1017/S0962492900002440
9. B. Fornberg, *A practical guide to pseudospectral methods*, Cambridge University Press, 1996. https://doi.org/10.1017/CBO9780511626357
10. J. S. Hesthaven, S. Gottlieb, D. Gottlieb, *Spectral methods for time-dependent problems*, Cambridge University Press, 2007. https://doi.org/10.1017/CBO9780511618352
11. C. Canuto, M. Y. Hussaini, A. Quarteroni, T. A. Zang, *Spectral methods: fundamentals in single domains*, Springer Berlin, Heidelberg, 2006. https://doi.org/10.1007/978-3-540-30726-6
12. D. Garg, *Advances in global pseudospectral methods for optimal control*, Ph.D. Thesis, Gainesville, FL: University of Florida, 2011.
13. G. T. Huntington, *Advancement and analysis of a Gauss pseudospectral transcription for optimal control problems*, Ph.D. Thesis, Massachusetts: MIT, 2007.
14. J. T. Betts, *Practical methods for optimal control using nonlinear programming*, Philadelphia, PA: SIAM, 2001.
15. C. L. Darby, *hp-Pseudospectral method for solving continuous-time nonlinear optimal control problems*, Ph.D. Thesis, Florida: University of Florida, 2011.
16. O. von Stryk, R. Bulirsch, Direct and indirect methods for trajectory optimization, *Ann. Oper. Res.*, **37** (1992), 357–373. https://doi.org/10.1007/BF02071065
17. C. C. Francolin, *Costate estimation for optimal control problems using orthogonal collocation at Gaussian quadrature points*, Ph.D. Thesis, Florida: University of Florida, 2013.

18. K. T. Elgindy, K. A. Smith-Miles, Fast, accurate, and small-scale direct trajectory optimization using a Gegenbauer transcription method, *J. Comput. Appl. Math.*, **251** (2013), 93–116. https://doi.org/10.1016/j.cam.2013.03.032

19. K. T. Elgindy, Gegenbauer collocation integration methods: advances in computational optimal control theory, *Bull. Aust. Math. Soc.*, **89** (2014), 168–170. https://doi.org/10.1017/S0004972713001044

20. K. T. Elgindy, K. A. Smith-Miles, Optimal Gegenbauer quadrature over arbitrary integration nodes, *J. Comput. Appl. Math.*, **242** (2013), 82–106. https://doi.org/10.1016/j.cam.2012.10.020

21. K. T. Elgindy, B. Karasözen, Distributed optimal control of viscous Burgers' equation via a high-order, linearization, integral, nodal discontinuous Gegenbauer-Galerkin method, *Optim. Control Appl. Methods*, **41** (2020), 253–277. https://doi.org/10.1002/oca.2541

22. C. J. Kim, S. Sung, A comparative study of transcription techniques for nonlinear optimal control problems using a pseudo-spectral method, *Int. J. Aeronaut. Space Sci.*, **16** (2015), 264–277. https://doi.org/10.5139/IJASS.2015.16.2.264

23. K. T. Elgindy, S. A. Dahy, High-order numerical solution of viscous Burgers' equation using a Cole-Hopf barycentric Gegenbauer integral pseudospectral method, *Math. Methods Appl. Sci.*, **41** (2018), 6226–6251. https://doi.org/10.1002/mma.5135

24. K. T. Elgindy, H. M. Refat, High-order shifted Gegenbauer integral pseudo-spectral method for solving differential equations of Lane–Emden type, *Appl. Numer. Math.*, **128**, (2018), 98–124. https://doi.org/10.1016/j.apnum.2018.01.018

25. C. W. Clenshaw, A. R. Curtis, A method for numerical integration on an automatic computer, *Numer. Math.*, **2** (1960), 197–205. https://doi.org/10.1007/bf01386223

26. S. E. El-Gendi, Chebyshev solution of differential, integral and integro-differential equations, *Comput. J.*, **12** (1969), 282–287. https://doi.org/10.1093/comjnl/12.3.282

27. X. Gao, T. Li, Q. Shan, Y. Xiao, L. Yuan, Y. Liu, Online optimal control for dynamic positioning of vessels via time-based adaptive dynamic programming, *J. Ambient Intell. Human. Comput.*, 2019, 1–13. https://doi.org/10.1007/s12652-019-01522-9

28. D. Wang, M. Ha, M. Zhao, The intelligent critic framework for advanced optimal control, *Artif. Intell. Rev.*, **55** (2022), 1–22. https://doi.org/10.1007/s10462-021-10118-9

29. B. Pang, L. Cui, Z. P. Jiang, Human motor learning is robust to control-dependent noise, *Biol. Cybern.*, **116** (2022), 307–325. https://doi.org/10.1007/s00422-022-00922-z

30. R. F. Baum, Existence theorems for Lagrange control problems with unbounded time domain, *J. Optim. Theory Appl.*, **19** (1976), 89–116. https://doi.org/10.1007/BF00934054

31. G. R. Bates, Lower closure and existence theorems for optimal control problems with infinite horizon, *J. Optim. Theory Appl.*, **24** (1978), 639–649. https://doi.org/10.1007/BF00935304

32. A. Haurie, Existence and global asymptotic stability of optimal trajectories for a class of infinite-horizon, nonconvex systems, *J. Optim. Theory Appl.*, **31** (1980), 515–533. https://doi.org/10.1007/BF00934475

33. D. A. Carlson, A. Haurie, *Infinite horizon optimal control: theory and applications*, Springer Berlin, Heidelberg, 1987. https://doi.org/10.1007/978-3-662-02529-1

34. E. J. Balder, An existence result for optimal economic growth problems, *J. Math. Anal. Appl.*, **95** (1983), 195–213. https://doi.org/10.1016/0022-247x(83)90143-9

35. D. A. Carlson, Existence of finitely optimal solutions for infinite-horizon optimal control problems, *J. Optim. Theory Appl.*, **51** (1986), 41–62. https://doi.org/10.1007/BF00938602

36. L. Wang, Existence and uniqueness of solutions for a class of infinite-horizon systems derived from optimal control, *Int. J. Math. Math. Sci.*, **2005** (2005), 837–843. https://doi.org/10.1155/IJMMS.2005.837

37. S. Pickenhain, Infinite horizon optimal control problems in the light of convex analysis in hilbert spaces, *Set-Valued Var. Anal.*, **23** (2015), 169–189. https://doi.org/10.1007/s11228-014-0304-5

38. K. O. Besov, On Balder's existence theorem for infinite-horizon optimal control problems, *Math. Notes*, **103** (2018), 167–174. https://doi.org/10.1134/s0001434618010182

39. A. V. Dmitruk, N. V. Kuz'kina, Existence theorem in the optimal control problem on an infinite time interval, *Math. Notes*, **78** (2005), 466–480. https://doi.org/10.1007/s11006-005-0147-3

40. S. M. Aseev, An existence result for infinite-horizon optimal control problem with unbounded set of control constraints, *IFAC-PapersOnLine*, **51** (2018), 281–285. https://doi.org/10.1016/j.ifacol.2018.11.396

41. V. Basco, H. Frankowska, Hamilton–Jacobi–Bellman equations with time-measurable data and infinite horizon, *Nonlinear Differ. Equ. Appl.*, **26** (2019), 7. https://doi.org/10.1007/s00030-019-0553-y

42. H. Halkin, Necessary conditions for optimal control problems with infinite horizons, *Econometrica*, **42** (1974), 267–272. https://doi.org/10.2307/1911976

43. D. Garg, W. Hager, A. V. Rao, Gauss pseudospectral method for solving infinite-horizon optimal control problems, In: *AIAA guidance, navigation, and control conference*, Toronto, Ontario, Canada: AIAA, 2012, 1–9. https://doi.org/10.2514/6.2010-7890

44. D. Garg, W. W. Hager, A. V. Rao, Pseudospectral methods for solving infinite-horizon optimal control problems, *Automatica*, **47** (2011), 829–837. https://doi.org/10.1016/j.automatica.2011.01.085

45. D. Garg, M. A. Patterson, C. Francolin, C. L. Darby, G. T. Huntington, W. W. Hager, et al., Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method, *Comput. Optim. Appl.*, **49** (2011), 335–358. https://doi.org/10.1007/s10589-009-9291-0

46. X. Tang, J. Chen, Direct trajectory optimization and costate estimation of infinite-horizon optimal control problems using collocation at the flipped Legendre-Gauss-Radau points, *IEEE/CAA J. Autom. Sin.*, **3** (2016), 174–183. https://doi.org/10.1109/JAS.2016.7451105

47. M. Shahini, M. A. Mehrpouya, Transformed Legendre spectral method for solving infinite horizon optimal control problems, *IMA J. Math. Control Inf.*, **35** (2018), 341–356. https://doi.org/10.1093/imamci/dnw051

48. D. Gottlieb, C. W. Shu, On the Gibbs phenomenon IV: recovering exponential accuracy in a subinterval from a Gegenbauer partial sum of a piecewise analytic function, *Math. Comput.*, **64** (1995), 1081–1095. https://doi.org/10.2307/2153484

49. D. Gottlieb, C. W. Shu, On the Gibbs phenomenon and its resolution, *SIAM Rev.*, **39** (1997), 644–668. https://doi.org/10.1137/S0036144596301390

50. J. R. Kamm, T. O. Williams, J. S. Brock, S. Li, Application of Gegenbauer polynomial expansions to mitigate Gibbs phenomenon in Fourier–Bessel series solutions of a dynamic sphere problem, *Int. J. Numer. Methods Biomed. Eng.*, **26** (2010), 1276–1292. https://doi.org/10.1002/cnm.1207

51. K. T. Elgindy, Optimal control of a parabolic distributed parameter system using a fully exponentially convergent barycentric shifted Gegenbauer integral pseudospectral method, *J. Ind. Manag. Optim.*, **14** (2018), 473–496. https://doi.org/10.3934/jimo.2017056

52. W. A. Light, A comparison between Chebyshev and ultraspherical expansions, *IMA J. Appl. Math.*, **21** (1978), 455–460. https://doi.org/10.1093/imamat/21.4.455

53. J. P. Boyd, Orthogonal rational functions on a semi-infinite interval, *J. Comput. Phys.*, **70** (1987), 63–88. https://doi.org/10.1016/0021-9991(87)90002-7

54. C. Canuto, M. Y. Hussaini, A. Quarteroni, T. A. Zang, *Spectral methods in fluid dynamics*, Springer Berlin, Heidelberg, 1988. https://doi.org/10.1007/978-3-642-84108-8

55. F. Fahroo, I. M. Ross, Pseudospectral methods for infinite-horizon nonlinear optimal control problems, *J. Guid. Control Dynam.*, **31** (2008), 927–936. https://doi.org/10.2514/1.33117

56. G. Szegö, *Orthogonal polynomials*, American Mathematical Society, 1939.

57. H. Wang, D. Huybrechs, S. Vandewalle, Explicit barycentric weights for polynomial interpolation in the roots or extrema of classical orthogonal polynomials, *Math. Comput.*, **83** (2014), 2893–2914. https://doi.org/10.1090/S0025-5718-2014-02821-4

58. K. T. Elgindy, High-order adaptive Gegenbauer integral spectral element method for solving non-linear optimal control problems, *Optimization*, **66** (2017), 811–836. https://doi.org/10.1080/02331934.2017.1298597

59. J. P. Berrut, Linear rational interpolation of continuous functions over an interval, In: W. Gautschi, *Mathematics of computation 1943–1993: a half-century of computational mathematics*, Proceedings of Symposia in Applied Mathematics, Vancouver, British Columbia: AMS, 1994, 261–264. https://doi.org/10.1090/psapm/048/1314853

60. J. P. Berrut, H. D. Mittelmann, Lebesgue constant minimizing linear rational interpolation of continuous functions over the interval, *Comput. Math. Appl.*, **33** (1997), 77–86. https://doi.org/10.1016/S0898-1221(97)00034-5

61. J. M. Carnicer, Weighted interpolation for equidistant nodes Carnicer, *Numer. Algor.*, **55** (2010), 223–232. https://doi.org/10.1007/s11075-010-9399-4

62. Q. Wang, P. Moin, G. Iaccarino, A rational interpolation scheme with superpolynomial rate of convergence, *SIAM J. Numer. Anal.*, **47** (2010), 4073–4097. https://doi.org/10.1137/080741574

63. L. Bos, S. De Marchi, K. Hormann, J. Sidon, Bounding the Lebesgue constant for Berrut's rational interpolant at general nodes, *J. Approx. Theory*, **169** (2013), 7–22. https://doi.org/10.1016/j.jat.2013.01.004

64. J. P. Berrut, Rational functions for guaranteed and experimentally well-conditioned global interpolation, *Comput. Math. Appl.*, **15** (1988), 1–16. https://doi.org/10.1016/0898-1221(88)90067-3

65. L. Bos, S. De Marchi, K. Hormann, On the Lebesgue constant of Berrut's rational interpolant at equidistant nodes, *J. Comput. Appl. Math.*, **236** (2011), 504–510. https://doi.org/10.1016/j.cam.2011.04.004

66. K. T. Elgindy, High-order, stable, and efficient pseudospectral method using barycentric Gegenbauer quadratures, *Appl. Numer. Math.*, **113** (2017), 1–25. https://doi.org/10.1016/j.apnum.2016.10.014

67. M. R. Hestenes, Multiplier and gradient methods, *J. Optim. Theory Appl.*, **4** (1969), 303–320. https://doi.org/10.1007/BF00927673

68. M. J. D. Powell, A method for nonlinear constraints in minimization problems, *Optimization*, 1969, 283–298.

69. K. T. Elgindy, Optimization via Chebyshev polynomials, *J. Appl. Math. Comput.*, **56** (2018), 317–349. https://doi.org/10.1007/s12190-016-1076-x

70. P. E. Murray, W. Murray, M. A. Saunders, SNOPT: an SQP algorithm for large-scale constrained optimization, *SIAM J. Optim.*, **12** (2002), 979–1006. https://doi.org/10.1137/S1052623499350013

71. P. E. Murray, W. Murray, M. A. Saunders, SNOPT: an SQP algorithm for large-scale constrained optimization, *SIAM Rev.*, **47** (2005), 99–131. https://doi.org/10.1137/S0036144504446096

72. D. E. Kirk, *Optimal control theory: an introduction*, Englewood Cliffs, N.J.: Prentice-Hall, 1970.

73. K. Mamehrashi, A. Nemati, A new approach for solving infinite horizon optimal control problems using Laguerre functions and Ritz spectral method, *Int. J. Comput. Math.*, **97** (2020), 1529–1544. https://doi.org/10.1080/00207160.2019.1628949

74. H. S. Nik, P. Rebelo, M. S. Zahedi, Solution of infinite horizon nonlinear optimal control problems by piecewise Adomian decomposition method, *Math. Model. Anal.*, **18** (2013), 543–560. https://doi.org/10.3846/13926292.2013.841598

Appendix

A. Barycentric GRDM

To construct the barycentric GRDM, we follow the derivation presented in [23] and multiply both sides of Eq (4.6) by $x - \tau_j \,\forall j$ to render them differentiable at $x = \tau_j$ such that

$$\mathcal{L}_{n,i}(x) \sum_{k=0}^{n} \frac{\xi_k \left(x - \tau_j\right)}{x - \tau_k} = \frac{\xi_i \left(x - \tau_j\right)}{x - \tau_i}, \quad i = 0, \ldots, n. \tag{A.1}$$

Letting $S(x) = \sum_{k=0}^{n} \frac{\xi_k \left(x - \tau_j\right)}{x - \tau_k}$ and differentiating Eq (A.1) with respect to x yields

$$S(x)\mathcal{L}'_{n,i}(x) + \mathcal{L}_{n,i}(x)S'(x) = \xi_i \left(\frac{x - \tau_j}{x - \tau_i}\right)', \quad i = 0, \ldots, n. \tag{A.2}$$

Since $S(\tau_j) = \xi_j, S'(\tau_j) = \sum_{j \neq k} \frac{\xi_k}{\tau_j - \tau_k}$ and $\mathcal{L}_{n,i}(\tau_j) = 0 \, \forall i \neq j$, the off-diagonal elements of the differentiation matrix $\mathbf{D} = (d_{j,i})_{0 \leq j,i \leq n}$ can be calculated by using the following formula:

$$d_{j,i} = \mathcal{L}'_{n,i}(\tau_j) = \frac{\xi_i/\xi_j}{\tau_j - \tau_i}, \quad \forall i \neq j. \tag{A.3}$$

For $i = j$, we have $\sum_{i=0}^{n} \mathcal{L}_{n,i}(x) = 1$, so $\sum_{i=0}^{n} \mathcal{L}'_{n,i}(x) = 0$ and

$$d_{i,i} = \mathcal{L}'_{n,i}(\tau_i) = -\sum_{j \neq i} \mathcal{L}'_{n,j}(\tau_i) = -\sum_{i \neq j} d_{i,j}, \quad i = 0, \ldots, n. \tag{A.4}$$

Hence, the derivative of a real-valued function $f \in C^1[-1,1]$ can be approximated at the GGR points by using the following formula:

$$f'(\tau_j) \approx \sum_{i=0}^{n} d_{j,i} f_i, \quad j = 0, \ldots, n. \tag{A.5}$$

B. Computational algorithms

Algorithm B.1 First switching formula of the barycentric weights for the GGR points.

```
Input: Positive integer $n$; a real number $\alpha > -1/2$; the set of GGR points and quadrature weights $\{\tau_i, \varpi_i\}_{i=0}^{n}$; a relatively small positive real number $\varepsilon$.
Output: Barycentric weights $\xi_i, i = 0, \ldots, n$.
$\xi_0 \leftarrow -\sqrt{(2\alpha+1)\varpi_0}$.
for $i = 1$ to $n$ do
    if $|1 - \tau_i| > \varepsilon$ then
        $\xi_i \leftarrow (-1)^{i-1} \sqrt{(1-\tau_i)\,\varpi_i}$.
    else
        $\xi_i \leftarrow (-1)^{i-1} \sin\left(\frac{1}{2}\cos^{-1}\tau_i\right)\sqrt{2\varpi_i}$.
    end if
end for
Stop.
```

Algorithm B.2 Second switching formula of the barycentric weights for the GGR points.

```
Input: Positive integer $n$; a real number $\alpha > -1/2$; the set of GGR points and quadrature weights $\{\tau_i, \varpi_i\}_{i=0}^{n}$; a relatively small positive real number $\varepsilon$.
Output: Barycentric weights $\xi_i, i = 0, \ldots, n$.
$\xi_0 \leftarrow -\sqrt{(2\alpha + 1)\varpi_0}$.
for $i = 1$ to $n$ do
    if $|1 - \tau_i| > \varepsilon$ then
        $\xi_i \leftarrow (-1)^{i-1} \sqrt{(1 - \tau_i)\, \varpi_i}$.
    else
        $\xi_i \leftarrow (-1)^{i-1} \sin\left(\cos^{-1} \tau_i\right) \sqrt{\frac{\varpi_i}{1 + \tau_i}}$.
    end if
end for
Stop.
```

AIMS Press